

According to New Revised Credit System Syllabus

SPPU

Third Year Degree Course In  
COMPUTER ENGINEERING (Sem - I)

# THEORY OF COMPUTATION

## Includes

- Model Question Papers For Practice  
(In Sem-30 Marks & End Sem- 70 Marks)



<xml /> NP

Mrs. SWATI K. BHAVSAR  
Ms. ABOLEE H. PATIL

Mrs. AKSHITA S. CHANCHLANI  
Mrs. NIKHITA A. PATIL(NERKAR)

[www.pragationline.com](http://www.pragationline.com)

 [www.facebook.com/nirallbooks](https://www.facebook.com/nirallbooks)

 **NIRALI**  
PRAKASHAN  
ADVANCEMENT OF KNOWLEDGE

A TEXT BOOK OF

# THEORY OF COMPUTATION

FOR  
SEMESTER – I  
THIRD YEAR DEGREE COURSE IN  
COMPUTER ENGINEERING

Strictly According to New Revised Credit System Syllabus  
of Savitribai Phule Pune University  
(w.e.f. June 2017)

**Mrs. SWATI K BHAVSAR**

ME (Comp Engg)  
Assistant Professor,  
Computer Engg. Deptt.,  
Matoshri College of Engg. & Research Centre  
Nashik.

**Mrs. AKSHITA S CHANCHLANI**

B.Tech. (VJTI) ME (IT)  
Assistant Professor,  
Computer Engg. Deptt.,  
Sinhgad Academy of Engineering,  
Kondhwa, Pune.

**Ms. ABOLEE H. PATIL**

ME (Comp.)  
Assistant Professor,  
Computer Engg. Deptt.,  
MIT College of Engineering,  
Kothrud, Pune.

**Mrs. NIKHITA A. PATIL (NERKAR)**

ME (Comp.)  
Assistant Professor,  
Computer Engg. Deptt.,  
RMD Sinhgad School of Engineering  
Warje, Pune.

Price ₹ 400.00

 **NIRALI**  
PRAKASHAN  
ADVANCEMENT OF KNOWLEDGE

**N4186**

**First Edition : June 2017****© : Authors**

The text of this publication, or any part thereof, should not be reproduced or transmitted in any form or stored in any computer storage system or device for distribution including photocopy, recording, taping or information retrieval system or reproduced on any disc, tape, perforated media or other information storage device etc., without the written permission of Authors with whom the rights are reserved. Breach of this condition is liable for legal action.

Every effort has been made to avoid errors or omissions in this publication. In spite of this, errors may have crept in. Any mistake, error or discrepancy so noted and shall be brought to our notice shall be taken care of in the next edition. It is notified that neither the publisher nor the authors or seller shall be responsible for any damage or loss of action to any one, of any kind, in any manner, therefrom.

**Published By :****Polyplate****Printed By :****NIRALI PRAKASHAN****YOGIRAJ PRINTERS AND BINDERS**

Abhyudaya Pragati, 1312, Shivaji Nagar,

Survey No. 10/1A, Ghule Industrial Estate

Off J.M. Road, Pune – 411005

Nanded Gaon Road

Tel - (020) 25512336/37/39, Fax - (020) 25511379

Nanded, Pune - 411041

Email : niralipune@pragationline.com

Mobile No. 9404233041/9850046517

**DISTRIBUTION CENTRES****PUNE**

**Nirali Prakashan** : 119, Budhwar Peth, Jogeshwari Mandir Lane, Pune 411002, Maharashtra  
Tel : (020) 2445 2044, 66022708, Fax : (020) 2445 1538

Email : bookorder@pragationline.com, niralilocal@pragationline.com

**Nirali Prakashan** : S. No. 28/27, Dhyari, Near Pari Company, Pune 411041

Tel : (020) 24690204 Fax : (020) 24690316

Email : dhyari@pragationline.com, bookorder@pragationline.com

**MUMBAI**

**Nirali Prakashan** : 385, S.V.P. Road, Rasdhara Co-op. Hsg. Society Ltd.,  
Girgaum, Mumbai 400004, Maharashtra

Tel : (022) 2385 6339 / 2386 9976, Fax : (022) 2386 9976

Email : niralimumbai@pragationline.com

**DISTRIBUTION BRANCHES****JALGAON**

**Nirali Prakashan** : 34, V. V. Golani Market, Navi Peth, Jalgaon 425001,  
Maharashtra, Tel : (0257) 222 0395, Mob : 94234 91860

**KOLHAPUR**

**Nirali Prakashan** : New Mahadvar Road, Kedar Plaza, 1<sup>st</sup> Floor Opp. IDBI Bank  
Kolhapur 416 012, Maharashtra. Mob : 9850046155

**NAGPUR**

**Pratibha Book Distributors:** Above Maratha Mandir, Shop No. 3, First Floor,  
Rani Jhanshi Square, Sitabuldi, Nagpur 440012, Maharashtra  
Tel : (0712) 254 7129

**DELHI**

**Nirali Prakashan** : 4593/21, Basement, Aggarwal Lane 15, Ansari Road, Daryaganj  
Near Times of India Building, New Delhi 110002  
Mob : 08505972553

**BENGALURU**

**Pragati Book House** : House No. 1, Sanjeevappa Lane, Avenue Road Cross,  
Opp. Rice Church, Bengaluru – 560002.  
Tel : (080) 64513344, 64513355, Mob : 9880582331, 9845021552  
Email: bharatsavla@yahoo.com

**CHENNAI**

**Pragati Books** : 9/1, Montieth Road, Behind Taas Mahal, Egmore,  
Chennai 600008 Tamil Nadu, Tel : (044) 6518 3535,  
Mob : 94440 01782 / 98450 21552 / 98805 82331,  
Email : bharatsavla@yahoo.com

**[niralipune@pragationline.com](mailto:niralipune@pragationline.com) | [www.pragationline.com](http://www.pragationline.com)****Also find us on  [www.facebook.com/niralibooks](http://www.facebook.com/niralibooks)**

*Dedicated to...*

**“Our Parents”**

*... Authors*



## PREFACE

It gives us great pleasure to present the book '**Theory of Computation**' for the students of Third Year Degree Course in Computer Engineering of the Savitribai Phule Pune University. This book is strictly as per the new revised syllabus 2015 Pattern with effect from the Academic Year (2017-18).

As per New Revised Examination Scheme which has been implemented from this academic year, In-semester assessment carries 30 marks over first three units and End Semester Examination carries 70 marks over entire syllabus out of which first three units will carry 20 marks and units 4, 5, 6 will carry 50 marks. The theory course will have 4 credits.

The book is written such that all the basic concepts are explained in simplified manner. It is presented in a more conceptual manner rather than mathematical, as required by the new examination system. It is our objective to keep the presentation systematic, consistent, intensive and clear through explanatory notes and figures. Main feature of this book is, **Complete Coverage** of the New Credit System Syllabus with large number of Worked Solved Examples, Exercises, **Model Question Papers of In Sem.** and **End Sem. Exams.**

We are sure that this book will cater to all needs of students for this subject.

We would like to our sincere thanks to **Management** and **Dr. G.K. Kharate** (Principal, MCOERC Nashik), **Dr. K.P. Patil** (Principal SAE Kondhwa), **Dr. C.B. Bangal** (Principal RMDSSOE Pune.), **Dr. M.S. Nagmode** (Principal, MITCOE Pune), **Prof. B.B. Gite** (HOD SAE Kondhwa), **Mrs. V.M. Lomte** (HOD Comp. Engg. Dept. RMDSSOE, Pune.) **Mrs. R.K. Bedi** (HOD Comp. Engg. Dept. MITCOE Pune), for their untiring support in our work.

Our Special thanks to **Dr. (Mrs.) Varsha H. Patil** Vice Principal, Professor and Head, Computer Engg. Deptt. MCOERC, Nashik) for her valuable guidance, timely advice and continuous encouragement.

We also take this opportunity to express our sincere thanks to Shri. Dineshbhai Furia, Shri. Jignesh Furia, Mrs. Nirali Verma, Shri. M. P. Munde and entire team of Nirali Prakashan namely Mrs. Deepali Lachake (Co-ordinator), who really have taken keen interest and untiring efforts in publishing this text.

The advice and suggestions of our esteemed readers to improve the text are most welcomed, and will be highly appreciated.

**Pune**

**Authors**



# SYLLABUS

## **Unit I : Formal Language Theory and Finite Automata**

**(08 Hrs)**

Introduction to Formal language, introduction to language translation logic, Essentials of translation, Alphabets and languages, Finite representation of language, Finite Automata (FA): An Informal Picture of FA, Finite State Machine (FSM), Language accepted by FA, Definition of Regular Language, Deterministic and Nondeterministic FA(DFA and NFA), epsilon- NFA, FA with output: Moore and Mealy machines -Definition, models, inter-conversion.

Case Study: FSM for vending machine, spell checker

## **Unit II : Regular Expressions (RE)**

**(07 Hrs)**

Introduction, Operators of RE, Building RE, Precedence of operators, Algebraic laws for RE, Conversions: NFA to DFA, RE to DFA Conversions: RE to DFA, DFA to RE Conversions: State/loop elimination, Arden's theorem Properties of Regular Languages: Pumping Lemma for Regular languages, Closure and Decision properties.

Case Study: RE in text search and replace

## **Unit III : Context Free Grammars (CFG) and Languages**

**(07 Hrs)**

Introduction, Regular Grammar, Context Free Grammar- Definition, Derivation, Language of grammar, sentential form, parse tree, inference, derivation, parse trees, ambiguity in grammar and Language- ambiguous Grammar, Simplification of CFG: Eliminating unit productions, useless production, useless symbols, and  $\epsilon$ -productions, Normal Forms- Chomsky normal form, Greibach normal form, Closure properties of CFL, Decision properties of CFL, Chomsky Hierarchy, Application of CFG: Parser, Markup languages, XML and Document Type Definitions.

Case Study- CFG for Palindromes, Parenthesis Match,

## **Unit IV : Turing Machines**

**(08 Hrs)**

Turing Machine Model, Representation of Turing Machines, Language Acceptability by Turing Machines, Design of TM, Description of TM, Techniques for TM Construction, Variants of Turing Machines, The Model of Linear Bounded Automata , TM & Type 0 grammars, TM's Halting Problem.

## **Unit V : Pushdown Automata**

**(07 Hrs)**

Basic Definitions, Equivalence of Acceptance by Finite State & Empty stack, PDA & Context Free Language, Equivalence of PDA and CFG, Parsing & PDA: Top-Down Parsing, Top-down Parsing Using Deterministic PDA, Bottom-up Parsing, Closure properties and Deterministic PDA.

## **Unit VI : Undecidability and Intractable Problems**

**(07 Hrs)**

A Language that is not recursively enumerable, An un-decidable problem that is RE, Post Correspondence Problem, The Classes P and NP : Problems Solvable in Polynomial Time, An Example: Kruskal's Algorithm, Nondeterministic Polynomial Time, An NP Example: The Traveling Salesman Problem, Polynomial-Time Reductions NP Complete Problems, An NP-Complete Problem: The Satisfiability Problem, Tractable and Intractable, Representing Satisfiability, Instances, NP Completeness of the SAT Problem, A Restricted Satisfiability Problem: Normal Forms for Boolean Expressions, Converting Expressions to CNF, The Problem of Independent Sets, The Node-Cover Problem.

# CONTENTS

<b>Unit I : Formal Language Theory and Finite Automata</b>	<b>1.1-1.120</b>
1.1 Introduction to Formal Language	1.1
1.2 Introduction to Language Translation Logic	1.1
1.3 Essentials of Translation	1.2
1.4 Alphabets and Languages	1.2
1.4.1 An Alphabet	1.2
1.4.2 Languages	1.3
1.5 Finite Representation of Language	1.3
1.5.1 Operations on Languages	1.4
1.6 Finite Automata: An Informal Picture of Finite Automata	1.5
1.7 Finite State Machine(FSM)	1.5
1.8 Language accepted by Finite Automata	1.6
1.9 Definition of Regular Language	1.7
1.9.1 Regular Set	1.9
1.10 Definition of Finite Automata	1.9
1.10.1 Representation	1.10
1.11 Deterministic Finite Automata (DFA)	1.15
1.11.1 Definition of a Deterministic Finite Automaton	1.15
1.11.2 How do a DFA Process Strings ?	1.15
1.11.3 Simple Notations for DFA's	1.17
1.11.4 Extending the Transition Function to Strings	1.18
1.11.5 The Language of DFA	1.19
1.12 Non-deterministic Finite Automata (NFA)	1.34
1.12.1 Definition of NFA	1.34
1.12.2 The Extended Transition Function	1.35
1.12.3 The Language of an NFA	1.36
1.12.4 Equivalence of Deterministic and Non-deterministic Finite Automata	1.39
1.12.5 FA with $\epsilon$ -moves	1.43
1.13 Conversion of NFA with $\epsilon$ to NFA without $\epsilon$	1.49
1.13.1 Indirect Method	1.49
1.13.2 Direct Method	1.52
1.14 Finite Automata with Output	1.55
1.14.1 Moore Machine	1.55
1.14.2 Mealy Machine	1.56
1.14.3 Inter Conversion	1.57
• Case Studies	1.109
• Exercise	1.110
• University Question Bank	1.114

<b>Unit II : Regular Expressions (RE)</b>		<b>2.1-2.92</b>
2.1	Introduction	2.1
2.2	Operators of Regular Expressions	2.1
2.3	Building Regular Expressions	2.2
2.4	Precedence of Operators	2.7
2.4.1	Characteristics of Regular Expressions	2.8
2.5	Algebraic laws of regular expressions	2.8
2.6	Conversions: NFA to DFA	2.11
2.6.1	Indirect Method	2.11
2.6.2	Direct Method	2.14
2.7	Conversion of Regular Expression to DFA	2.16
2.7.1	Regular Expressions and Finite Automata	2.16
2.7.2	Regular Expression to DFA	2.17
2.7.3	RE to DFA (Indirect Method)	2.18
2.7.4	RE to DFA (Direct Method)	2.22
2.8	DFA to Regular Expressions conversions	2.26
2.8.1	State/ Loop Elimination	2.30
2.8.2	Arden's Theorem	2.32
2.9	Properties of regular languages	2.36
2.9.1	Pumping Lemma Proving Languages not to be Regular	2.36
2.9.2	Closure Properties of Regular Languages	2.39
2.9.3	Decision Properties of Regular Languages	2.40
•	Case Study	2.41
•	Exercise	2.81
•	University Question Bank	2.83
<b>Unit III : Context Free Grammars (CFG) and Languages</b>		<b>3.1-3.100</b>
3.1	Introduction to Grammar	3.1
3.1.1	Phrase Structure Grammar	3.1
3.1.2	Language Generated by the Grammar	3.2
3.2	Regular Grammar	3.4
3.2.1	Definition	3.4
3.2.2	Regular Grammar and Finite Automata	3.4
3.2.3	Construction of Regular Grammar for a Given FA	3.4
3.2.4	Construction of FA from a given Right Linear Grammar	3.5
3.2.5	Inter-conversion Between Left Linear GL and Right Linear GR Regular Grammar	3.7
3.2.6	Construction of FA from a Left Linear Grammar	3.8
3.3	The Context Free Grammar	3.17
3.3.1	Definition	3.17
3.3.2	Derivations	3.18

3.3.3	Language of grammar	3.19
3.3.4	Sentential Form	3.20
3.3.5	Derivation Tree and Derivations	3.21
3.3.6	Parse Tree	3.22
3.4	Ambiguity in grammar and language	3.25
3.5	Simplification of Context Free Grammar	3.30
3.5.1	Elimination of Useless Symbols	3.30
3.5.2	Computing the Generating and Reachable Symbols	3.32
3.5.3	Eliminating $\epsilon$ -productions	3.32
3.5.4	Decision Properties of CFL	3.33
3.5.5	Eliminating the Unit Productions	3.35
3.6	Normal forms	3.36
3.6.2	Greibach Normal Form (GNF)	3.37
3.6.3	Closure Properties of CFL	3.53
3.6.4	Decision properties of context free language	3.51
3.7	The Chomsky Hierarchy	3.58
3.8	Applications of CFG	3.60
3.8.1	Parser	3.60
3.8.2	Yacc	3.61
3.8.3	Markup Languages	3.76
3.8.4	XML and Document Type Definition	3.76
•	Case Studies	3.80
•	Exercise	3.91
•	University Question Bank	3.92
<b>Unit IV : Turing Machine</b>		<b>4.1-4.62</b>
4.1	Turing Machine Model	4.1
4.2	Representation of Turing Machine	4.2
4.3	Language Acceptability by Turing Machines	4.3
4.4	Design of Turing Machine	4.3
4.5	Description of the Turing Machine	4.4
4.6	Techniques for Turing Machine Construction	4.5
4.6.1	Storage in the State	4.5
4.6.2	Multiple Tracks	4.6
4.6.3	Subroutines	4.6
4.7	Variants of Turing Machine	4.6
4.7.1	Multitape Turing Machines	4.7
4.7.2	Non-deterministic Turing Machines	4.8
4.8	The Model of Linear Bounded Automata	4.9
4.9	TM and Type of Grammar	4.9

4.10	The Halting Problem	4.10
4.10.1	Consequences of the Halting Problem	4.11
4.11	Turing Machine and computers	4.12
4.11.1	Simulating a Turing Machine by Computer	4.12
4.11.2	Simulating a Computer by a Turing Machine	4.13
•	Exercise	4.58
•	University Question Bank	4.61
<b>Unit 5 : Pushdown Automata (PDA)</b>		<b>5.1-5.56</b>
5.1	Introduction	
5.2	Basic Definition	5.1
5.2.1	Definition	5.1
5.2.2	A Graphical Notation for PDA's	5.3
5.2.2	A Graphical Notation for PDA's	5.3
5.2.3	Instantaneous Description of a PDA	5.4
5.3	Equivalence of Acceptance By Finite State And Empty Stack	5.5
5.3.1	Acceptance by Final State	5.5
5.3.2	Acceptance by Empty Stack	5.6
5.3.3	Pictorial representation of PDA	5.6
5.3.4	Non-deterministic Pushdown Automata (NPDA)	5.13
5.4	Pushdown Automata And Context Free Language	5.16
5.5	Equivalence of PDA's and CFG's	5.16
5.5.1	CFG to PDA	5.17
5.5.2	Construction of (pictorial) PDA from CFG	5.17
5.5.3	From PDA to CFG	5.20
5.6	Parsing and PDA	5.23
5.6.1	Top Down Parsing	5.24
5.6.2	Bottom up Parser	5.24
5.7.	Closure properties	5.24
5.8	Deterministic pushdown automaton (DPDA)	5.24
5.8.1	Definition of DPDA	5.25
5.8.2	Regular Language and DPDA	5.26
5.8.3	DPDA's and CFLs	5.27
5.8.4	DPDA's and Ambiguous Grammar	5.27
•	Exercise	5.50
•	University Question Bank	5.51

**Unit VI : Undecidability and Intractable Problems** **6.1-6.26**

6.1	Introduction	6.1
6.2	A Language that is not Recursively Enumerable	6.1
6.3	An Undecidable Problem that is Recursively Enumerable	6.2
6.3.1	Recursive Languages	6.2
6.3.2	Complements of Recursive and RE Languages	6.3
6.3.3	The Universal Language	6.4
6.4	Post Correspondence Problem (PCP)	6.5
6.4.1	A Modified PCP (MPCP)	6.7
6.5	The classes P and NP	6.9
6.5.1	Problems Solvable in Polynomial Time	6.9
6.5.2	An Example : Kruskal's Algorithm	6.9
6.5.3	Non-deterministic Polynomial Time	6.10
6.5.4	An NP Example: Travelling Salesman Problem	6.10
6.5.5	Polynomial Time Reductions	6.12
6.6	NP Complete Problems	6.13
6.6.1	The Satisfiability Problems	6.13
6.6.2	Tractable and Intractable	6.14
6.6.3	Representing SAT Instances	6.14
6.6.4	NP Completeness of SAT Problems	6.15
6.7	A Restricted satisfiability problem	6.15
6.7.1	Normal Forms for Boolean Expression	6.15
6.7.2	Converting Expressions to CNF	6.16
6.7.3	The Problem of Independent Sets	6.16
6.7.4	Node Cover Problem	6.16
•	Exercise	6.25
•	<b>Model Question Papers</b>	
➤	<b>In-Sem 30 Marks</b>	P.1-P.2
➤	<b>End-Sem 70 Marks</b>	P.2-P.4



## Unit I

# FORMAL LANGUAGE THEORY AND FINITE AUTOMATA

---

## 1.1 INTRODUCTION TO FORMAL LANGUAGE

- A language is a subset of all possible strings formed from a given set of symbols.
- In general, a natural language is one that has evolved over time for the purpose of human communication.
- Examples include English, German, and Hindi. Such languages continue to evolve without regard for formal grammatical rules.
- Natural languages rarely conform to simple or obvious a grammatical rules.
- In contrast to natural languages, formal languages are defined by, and hence rigorously conform to, pre established rules.
- Examples mathematical languages such as algebra and propositional logic. It is because of this adherence to rules that the construction of efficient computerized translators for programming languages is possible, whereas it is the lack of adherence to established rules that makes it difficult to construct a translator for a natural language.

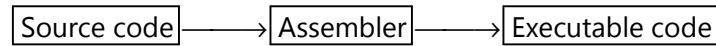
**Alphabet :** A non-empty, finite set of symbols from which the strings to be analyzed are constructed, denoted as  $\Sigma$ .

**Language :** A subset of  $\Sigma^*$  (collection of all strings of finite length including the empty string that can be constructed alphabet  $\Sigma$  being used) is called as language over  $\Sigma$ .

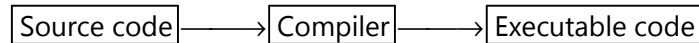
## 1.2 INTRODUCTION TO LANGUAGE TRANSLATION LOGIC

- A translator is a computer program that performs translation of a program written in a given programming language into a functionally equivalent program in a different ways computer language.
- But while translating it takes care that it should not loose the functional or logical structure of original code.
- It mainly consist of translation between high level languages e.g. C, C++, Java ink low level languages such as, machine code, assembly language and byte code.
- As the computer can only understand programs defined using machine code.
- There are three types of system s/w are used for translating the code which is written by programmer into a form which is understandable to machine are :

**1. Assembler :** It converts an assembly program into machine language.



**2. Compiler :** It translates a high level program into machine code.



**3. Interpreter:** It translates high level source code into executable code.

The difference between compiler and interpreter is that, it translates one line at a time and then executes it.

### 1.3 ESSENTIALS OF TRANSLATION

- A translator is very essential for a program which is written in a given programming language and to be converted into a functionally equivalent program.
- It should work in efficient way without losing functional or logical structure of original code.
- As computer does not understand the high level language, there is a necessity of translator to translate into machine level language.
- Semantic plays an important role for language translation. It is associated with meaning of language.

### 1.4 ALPHABETS AND LANGUAGES

#### 1.4.1 An Alphabet

- An alphabet is a finite non-empty set of symbols. It is denoted by  $\Sigma$ . Symbols may be numbers or letters.
- For example,
  - 1)  $\Sigma = \{0, 1\}$
  - 2)  $\Sigma = \{a, b, c, \dots, z\}$
  - 3)  $\Sigma = \{0, 1, 2, \dots, 9\}$

#### Power of an alphabet :

- If  $\Sigma$  is an alphabet, the set of all strings of a certain length 'k' from that alphabet is nothing but power of alphabet.

For example,

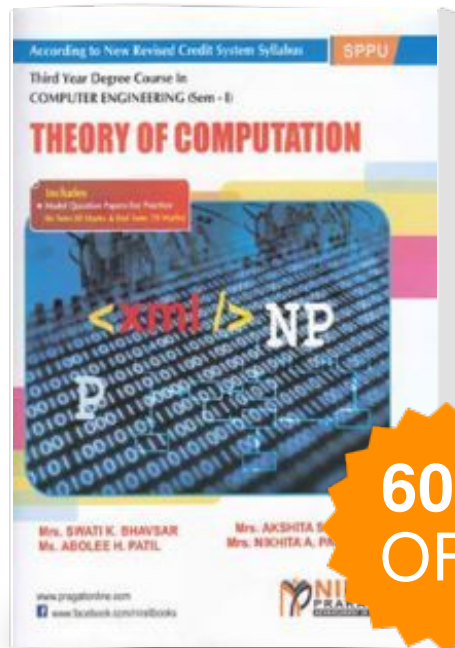
$$\Sigma^0 = \{\epsilon\}$$

$$\Sigma = \{0, 1\}$$

$$\Sigma^1 = \{0, 1\}$$

$$\Sigma^2 = \{00, 11, 01, 10\}$$

# Theory Of Computation



60%  
OFF

Publisher : Niralı Prakashan

ISBN : 9789386700087

Author : Mrs. Swati K  
Bhavsar, Mrs. Akshita S  
Chanchlani, Ms. Abolees H.  
Patil, Mrs. Nikhita A. Patil  
(Nerkar)

Type the URL : <http://www.kopykitab.com/product/20812>



Get this eBook