Contents

Contents **vii**

Introduction **ix**

1 Preliminaries **1**
1.1 Alphabets and languages 1
1.2 Multisets 3
1.3 Graphs and trees 4
1.4 Recursive functions 5
1.5 Diophantine sets 8
1.6 Turing machines 8
1.7 Complexity classes 11

I Computational completeness **13**

2 Cellular computing systems with membranes **15**
2.1 From cell to P systems 15
2.2 Formal scheme of a P system 21

3 P systems with external output **25**
3.1 Membrane structures 25
3.2 Syntax of P systems with external output 26
3.3 Semantics of P systems with external output 29
3.4 Variants of P systems with external output 36

4 P systems as accepting devices **37**
4.1 Language accepting P systems 37
4.2 Computational completeness through Turing machines . . . 39
4.3 Formal verification 51

5 P systems as language generating devices **75**
5.1 Language generating P systems 75

5.2 Computational completeness through formal languages . . . 77
5.3 Formal verification 85

6 P systems as computing devices 97
6.1 Function computing P systems 97
6.2 Computational completeness through recursive functions . . 99

7 P systems as set generating devices 115
7.1 Set generating P systems 115
7.2 Computational completeness through Diophantine sets . . . 116

II Computational complexity 125

8 A Complexity Theory in P Systems 127
8.1 Preliminary definitions 127
8.2 Complexity classes in P systems 129

9 P Systems and the P = NP problem 139
9.1 Simulation of Turing machines by P systems 140
9.2 Simulation of P systems by Turing machines 148
9.3 Characterization of the **P** $\neq$ **NP** relation through P systems 156

10 P systems with active membranes 159
10.1 Syntax of P systems with active membranes 159
10.2 Semantics of P systems with active membranes 161

11 Linear solutions of the SAT and VALIDITY problems in systems with active membranes 171
11.1 P systems with active membranes as accepting devices . . . 172
11.2 Resolution of **SAT** problem 174
11.3 Resolution of **VALIDITY** problem 193

Conclusions and future work 205

Bibliography 209