Contents

1 Formal languages, grammars, and their classification 7

1.1 Formal languages . . . 7

1.2 Classification of languages (the Chomsky hierarchy) . . . 9

1.3 Grammars and their classification . . . 11

2 Regular languages 15

2.1 Regular grammars . . . 15

2.1.1 Designing a regular grammar for a given language . . . 15

2.2 Finite automata . . . 22

2.2.1 Designing a finite automaton for a given language . . . 25

2.2.2 Removal of unreachable and redundant states . . . 37

2.2.3 Removal of ε-transitions . . . 41

2.2.4 Transformation of NFA with multiple initial states to NFA with one initial state . 44

2.2.5 Determinization . . . 45

2.2.6 DFA minimization . . . 49

2.2.7 Operations on finite automata . . . 56

2.3 Regular expressions . . . 63

2.3.1 Designing a regular expression for a given language . . . 63

2.3.2 Simplification of regular expressions . . . 65

2.3.3 Regular equations . . . 66

2.3.4 A standard system of regular equations . . . 67

2.3.5 Derivatives of regular expressions . . . 70

2.4 Relations between regular grammars, regular expressions, and finite automata . . . 71

2.4.1 Regular grammar → finite automaton . . . 72

2.4.2 Finite automaton → regular grammar . . . 73

2.4.3 Regular expression → finite automaton . . . 75

2.4.4 Finite automaton → regular expression . . . 82

2.4.5 Regular grammar → regular expression 87
2.4.6 Regular expression → regular grammar 91
2.5 Pumping lemma 93
2.5.1 Using pumping lemma for proving that a given language is not regular 94
2.5.2 Determination of pumping lemma constant for a given language 113
2.6 Myhill–Nerode theorem 116
2.6.1 Using Myhill-Nerode theorem for proving that a given language is not regular . . 117
2.6.2 Using Myhill-Nerode theorem for proving that a given language is regular 124
2.7 Closure properties 125
2.7.1 Using closure properties for proving that a given language is not regular 125

3 Context-free languages **128**
3.1 Context-free grammars 128
3.1.1 Parse tree 128
3.1.2 Ambiguous and unambiguous context-free grammars 129
3.1.3 Designing a context-free grammar for a given language 130
3.1.4 Exclusion of ε-rules 141
3.1.5 Exclusion of simple rules 143
3.1.6 Exclusion of redundant symbols 145
3.1.7 Exclusion of left recursion 148
3.1.8 Chomsky normal form 151
3.2 Cocke-Younger-Kasami algorithm 154
3.3 Pushdown automata 159
3.3.1 Designing a pushdown automaton for a given language 163
3.3.2 Parsing (syntactic analysis, context-free grammar → pushdown automaton) . . . 175

4 Formal translations **178**
4.1 Translation grammars 178
4.1.1 Designing a regular translation grammar for a given translation 179
4.1.2 Designing a context-free translation grammar for a given translation 180
4.2 Translation automata 183
4.2.1 Designing a finite transducer for a given translation 185
4.2.2 Designing a pushdown translation automaton for a given translation 186

5 Context-sensitive languages **190**
5.1 Context-sensitive grammars 190
5.1.1 Designing a context-sensitive grammar for a given language 191
5.2 Linear bounded automata 192
5.2.1 Designing a linear bounded automaton for a given language 194

6 Recursively enumerable languages **196**

6.1 Unrestricted grammars 196

6.1.1 Designing an unrestricted grammar for a given language 197

6.2 Turing machine 197

6.2.1 Designing a Turing machine for a given language 198

6.2.2 P and NP complexity classes 198