

Contents

1.	What are Regexes and Grammars?	1
1.1	Use Cases	1
1.2	Regexes or Regular Expressions?	3
1.3	What's So Special about Perl 6 Regexes?	3
2.	Getting Started with Perl 6	5
2.1	Installing Rakudo Perl 6	5
2.2	Using Rakudo Perl 6	6
2.3	Obtaining the Code Examples	7
2.4	Summary	8
3.	The Building Blocks of Regexes	9
3.1	Literals	9
3.2	Meta Characters vs. Literals	9
3.3	Anchors	10
3.4	Pre-Defined Character Classes	11
3.5	Quantifiers	14
3.6	Disjunction	16
3.7	Conjunction	17
3.8	Zero-Width Assertions	17
3.9	Summary	18
4.	Regexes and Perl 6 Code	19
4.1	Smart-Matching	19
4.2	Quote Forms	19
4.3	Modifiers	20
4.4	Comb and Split	22
4.5	Substitution	23
4.6	Crossing the Code and Regex Boundary	24
4.7	Summary	27
5.	Extracting Data from Regex Matches	28
5.1	Positional Captures	28
5.2	The Match Object	28
5.3	Named Captures	30
5.4	Backreferences	31
5.5	Match Objects Revisited	33

CONTENTS

5.6	Summary	33
6.	Regex Mechanics	34
6.1	Matching with State Machines	34
6.2	Regex Control Flow	40
6.3	Backtracking	40
6.4	Why Would You Want to Avoid Backtracking?	42
6.5	Frugal Quantifiers and Backtracking	44
6.6	Longest Token Matching	45
6.7	Summary	46
7.	Regex Techniques	47
7.1	Know your Data Format	47
7.2	Think about Invalid Inputs	49
7.3	Use Anchors	49
7.4	Matching Quoted Strings	49
7.5	Testing Regexes	52
7.6	Summary	56
8.	Reusing and Composing Regexes	57
8.1	Named Regexes	57
8.2	Whitespace	60
8.3	Grammars	62
8.4	Code Reuse with Grammars	63
8.5	Proto Regexes	65
8.6	Summary	69
9.	Parsing With Grammars	70
9.1	Understanding Grammars	70
9.2	Starting Simple	73
9.3	Assembling Complete Grammars	75
9.4	Debugging Grammars	75
9.5	Parsing Whitespace and Comments	78
9.6	Keeping State	80
9.7	Summary	87
10.	Extracting Data From Matches	88
10.1	Action Objects	88
10.2	Building ASTs with Action Objects	91
10.3	Keeping State in Action Objects	93
10.4	Summary	95
11.	Generating Good Parse Error Messages	96
11.1	Exploring The Problem	96
11.2	Assertions	97
11.3	Improved Position Reporting	98
11.4	High-Water Marks	99

CONTENTS

11.5	Parser Combinator and FAILGOAL	101
11.6	Which Techniques To Use?	102
11.7	Summary	103
12.	Unicode and Natural Language	104
12.1	Writing Systems	104
12.2	Bytes, Code Points, Graphemes, and Glyphs	105
12.3	Unicode Properties	106
12.4	Summary	107
13.	Case Studies	108
13.1	S-Expressions	108
13.2	Mathematical Expressions and Operator Precedence Parsers	114
13.3	Pythonesque, an Indentation-Based Language	120
13.4	Summary	127
14.	Acknowledgements	128