

Contents

<i>Preface</i>	<i>page xi</i>
<i>Acknowledgments</i>	<i>xiv</i>
1 An introduction to rippling	1
1.1 Overview	1
1.1.1 The problem of automating reasoning	1
1.1.2 Applications to formal methods	2
1.1.3 Proof planning and how it helps	3
1.1.4 Rippling: a common pattern of reasoning	3
1.2 A logical calculus of rewriting	5
1.3 Annotating formulas	8
1.4 A simple example of rippling	9
1.5 Using the given: fertilization	12
1.6 Rewriting with wave-rules	12
1.7 The preconditions of rippling	14
1.8 The bi-directionality of rippling	15
1.9 Proofs by mathematical induction	16
1.9.1 Recursive data types	17
1.9.2 Varieties of induction rule	18
1.9.3 Rippling in inductive proofs	20
1.10 The history of rippling	21
2 Varieties of rippling	24
2.1 Compound wave-fronts	24
2.1.1 An example of wave-front splitting	24
2.1.2 Maximally split normal form	25
2.1.3 A promise redeemed	26
2.1.4 Meta-rippling	27

viii	<i>Contents</i>	
2.1.5	Unblocking rippling with simplification	27
2.2	Rippling sideways and inwards	28
2.2.1	An example of sideways rippling	29
2.2.2	Universal variables in inductive proofs	31
2.2.3	Revised preconditions for rippling	32
2.2.4	Cancellation of inwards and outwards wave-fronts	32
2.3	Rippling-in after weak fertilization	33
2.3.1	An abstract example	33
2.3.2	Another way to look at it	34
2.3.3	A concrete example	35
2.4	Rippling towards several givens	36
2.4.1	An example using trees	36
2.4.2	Shaken but not stirred	38
2.4.3	Weakening wave-fronts	39
2.5	Conditional wave-rules	41
2.6	Rippling wave-fronts from given to goal	44
2.7	Higher-order rippling	45
2.8	Rippling to prove equations	48
2.9	Relational rippling	50
2.10	Summary	53
3	Productive use of failure	54
3.1	Eureka steps	54
3.2	Precondition analysis and proof patching	56
3.3	Revising inductions	58
3.3.1	Failure analysis	59
3.3.2	Patch: wave-front speculation	60
3.4	Lemma discovery	61
3.4.1	Failure analysis	61
3.4.2	Patch: lemma speculation	62
3.4.3	Alternative lemmas	64
3.4.4	Patch: lemma calculation	66
3.5	Generalizing conjectures	67
3.5.1	Failure analysis	67
3.5.2	Patch: sink speculation	68
3.5.3	Alternative generalizations	70
3.6	Case analysis	73
3.6.1	Failure analysis	73
3.6.2	Patch: casesplit calculation	73
3.7	Rotate length conjecture revisited	74

<i>Contents</i>		ix
3.7.1	Rotate length: conjecture generalization	74
3.7.2	Rotate length: lemma discovery	76
3.7.3	An automated reasoning challenge	77
3.8	Implementation and results	78
3.9	Summary	81
4	A formal account of rippling	82
4.1	General preliminaries	82
4.1.1	Terms and positions	83
4.1.2	Substitution and rewriting	84
4.1.3	Notation	85
4.2	Properties of rippling	86
4.2.1	Preliminaries	86
4.2.2	Properties of rippling	87
4.3	Implementing rippling: generate-and-test	88
4.3.1	Embeddings	89
4.3.2	Annotated terms and rippling	90
4.3.3	Implementation	91
4.4	Term annotation	92
4.4.1	The role of annotation	92
4.4.2	Formal definitions	93
4.5	Structure-preserving rules	96
4.6	Rippling using wave-rules	96
4.6.1	Why ordinary rewriting is not enough	97
4.6.2	Ground rippling	98
4.6.3	Annotated matching	100
4.6.4	(Non-ground) rippling	103
4.6.5	Termination	104
4.7	Orders on annotated terms	105
4.7.1	Simple annotation	105
4.7.2	Multi-hole annotation	108
4.7.3	Termination under $\succ^*$	110
4.8	Implementing rippling	113
4.8.1	Dynamic wave-rule parsing	114
4.8.2	Sinks and colors	116
5	The scope and limitations of rippling	118
5.1	Hit: bi-directionality in list reversal	118
5.2	Hit: bi-conditional decision procedure	120
5.3	Hit: reasoning about imperative programs	120

x	<i>Contents</i>	
5.4	Hit: $\lim+$ theorem	121
5.5	Hit: summing the binomial series	122
5.6	Hit: meta-logical reasoning	123
5.7	Hit: SAM's lemma	123
5.8	What counts as a failure?	124
5.9	Miss: mutual recursion	125
5.10	Miss: commuted skeletons	126
5.11	Miss: holeless wave-fronts	127
5.12	Miss: inverting a tower	129
5.13	Miss: difference removal	131
5.14	Best-first rippling	132
5.15	Rippling implementations and practical experience	133
5.16	Summary	134
6	From rippling to a general methodology	144
6.1	A general-purpose annotation language	146
6.2	Encoding constraints in proof search: some examples	148
6.2.1	Example 1: encoding rippling and difference reduction	148
6.2.2	Example 2: encoding basic ordered paramodulation and basic superposition	150
6.2.3	Example 3: Encoding window inference	154
6.2.4	Example 4: Proving theorems by reuse	156
6.2.5	Summary	159
6.3	Using annotations to implement abstractions	160
6.3.1	Limitations of abstractions	160
6.3.2	Abstractions on annotated terms	162
6.3.3	Examples revisited	165
6.4	Implementation	172
6.5	Summary	173
7	Conclusions	175
<i>Appendix 1</i>	<i>An annotated calculus and a unification algorithm</i>	<i>177</i>
A1.1	An annotation calculus	177
A1.2	Unification algorithm	183
A1.2.1	Soundness and completeness	188
<i>Appendix 2</i>	<i>Definitions of functions used in this book</i>	<i>190</i>
	<i>References</i>	<i>193</i>
	<i>Index</i>	<i>200</i>