

Contents

1. Introduction	1
1.1 Formal Methods	2
1.2 Formal Methods Tools	3
1.3 Inference Systems	5
1.3.1 Model Checkers	6
1.3.2 Interactive Theorem Provers	7
1.3.3 Automated Theorem Provers	8
2. Formal Methods in Software Engineering	11
2.1 Introduction	11
2.2 Formal Methods and the Software Life Cycle	12
2.2.1 The Waterfall Model	12
2.2.2 An Iterative Life Cycle	13
2.2.3 Refinements and Formal Methods	14
2.3 Degree and Scope of Formal Methods	15
2.4 Specification Languages and Proof Tasks	16
2.5 Current Situation	18
3. Processing of Logic	23
3.1 Automated Deduction	23
3.1.1 A First-Order Language	24
3.1.2 Model Theory	25
3.1.3 Formal Systems	27
3.2 Theorem Proving	28
3.2.1 Introduction	28
3.2.2 Clausal Normal Form	28
3.2.3 Resolution	30
3.2.4 Model Elimination	31
3.3 The Automated Prover SETHEO	32
3.3.1 Calculus and Proof Procedure	33
3.3.2 Extensions	34
3.3.3 System Architecture and Implementation	38
3.4 Common Characteristics of Automated Provers	40

4. Characteristics of Proof Tasks	43
4.1 Seen from the Outside	44
4.1.1 Number of Proof Tasks	44
4.1.2 Frequency	44
4.1.3 Size and Syntactic Richness	45
4.2 Logic-related Characteristics	48
4.2.1 Logic and Language	48
4.2.2 Simplification	49
4.2.3 Sorts and Types	49
4.2.4 Equality and Other Theories	51
4.3 Validity and Related Characteristics	52
4.3.1 Theorems vs. Non-theorems	52
4.3.2 Complexity	52
4.3.3 Expected Answer	55
4.3.4 What Is the Answer Worth?	55
4.3.5 Semantic Information	56
4.4 System-related Characteristics	57
4.4.1 Flow of Data and Control	57
4.4.2 Relation Between Proof Tasks	57
4.4.3 Automatic vs. Manual Generation	57
4.4.4 Guidance and Resource Limits	60
4.4.5 Human-understandable Proof Tasks	60
4.5 Discussion	61
4.6 Summary and Evaluation Form	61
4.7 Examples	63
4.7.1 Amphion	63
4.7.2 ILF	67
4.7.3 KIV and Automated Theorem Provers	68
5. Requirements	71
5.1 General Issues	71
5.1.1 Expressiveness	71
5.1.2 Soundness and Completeness	74
5.1.3 Can You Trust an ATP?	74
5.1.4 Proving and Other AI Techniques	75
5.2 Connecting the ATP	76
5.2.1 Reading and Preparing the Proof Tasks	77
5.2.2 Starting the Prover	79
5.2.3 Stopping the Prover	80
5.2.4 Analyzing the Results and Cleaning Up	80
5.3 Induction	81
5.4 Equality and Arithmetic	82
5.4.1 Handling of Equality	82
5.4.2 Arithmetic	82
5.4.3 Constraints	84

5.5	Logic Simplification	85
5.6	Sorts	85
5.7	Generation and Selection of Axioms	87
5.8	Handling of Non-Theorems	90
5.9	Control	91
5.9.1	Size of the Search Space	93
5.9.2	Influence of Parameters	94
5.9.3	Influence of the Formula	96
5.10	Additional Requirements	96
5.10.1	Software Engineering Requirements	96
5.10.2	Documentation and Support	98
6.	Case Studies	99
6.1	Verification of Authentication Protocols	101
6.1.1	Introduction	101
6.1.2	The Application	104
6.1.3	Using the Automated Prover	108
6.1.4	Experiments and Results	113
6.1.5	Assessment and Discussion	114
6.2	Verification of a Communication Protocol	114
6.2.1	Introduction	114
6.2.2	The Application	114
6.2.3	Using the Automated Prover	118
6.2.4	Experiments and Results	121
6.2.5	Assessment and Discussion	121
6.3	Logic-based Component Retrieval	122
6.3.1	Introduction	122
6.3.2	The Application	123
6.3.3	The Proof Tasks	128
6.3.4	Using the Automated Prover	129
6.3.5	Experiments and Results	132
6.3.6	Assessment and Discussion	135
7.	Specific Techniques for ATP Applications	137
7.1	Overview	138
7.1.1	Translation Phase	139
7.1.2	Preprocessing Phase	139
7.1.3	Execution Phase: Running the ATP	141
7.1.4	Postprocessing Phase	141
7.2	Handling of Non-First-Order Logics	141
7.2.1	Induction	142
7.2.2	Modal and Temporal Logics	146
7.3	Simplification	150
7.3.1	Static Simplification	151
7.3.2	Semantic Simplification	153

7.4	Sorts	156
7.4.1	Sorts as Unary Predicates	157
7.4.2	Sorted Unification	157
7.4.3	Compilation into Terms	158
7.5	Handling Non-Theorems	160
7.5.1	Detection of Non-Theorems by Simplification	162
7.5.2	Generation of Counter-Examples	163
7.5.3	A Generative Approach	167
7.6	Control	170
7.6.1	Combination of Search Paradigms	170
7.6.2	Parallel Execution	174
7.7	Postprocessing	185
7.7.1	Proof-related Information	185
7.7.2	Machine-oriented Proofs	189
7.7.3	Machine-oriented Proofs on the Source Level	189
7.7.4	Human-readable Proofs	190
7.8	Pragmatic Considerations	194
7.8.1	Input and Output Language	194
7.8.2	Software Engineering Issues	195
8.	Conclusions	197
8.1	Summary	197
8.2	Questions and Answers	199
	References	203
	Index	221



<http://www.springer.com/978-3-540-67989-9>

Automated Theorem Proving in Software Engineering

Schumann, J.M.

2001, XIV, 228 p., Hardcover

ISBN: 978-3-540-67989-9