

Preface	xvii
Acknowledgments	xix
1 Introduction	1
2 Logical Preliminaries	5
2.1 First-Order Logic	5
2.1.1 Syntax	5
2.1.2 Semantics	6
2.1.3 Soundness and Completeness	8
2.1.4 Many-Sorted First-Order Languages	8
2.1.5 Reducing Many-Sorted Logic to Standard Logic	10
2.1.6 Some Useful First-Order Inference Rules	10
2.1.7 A Limitation of First-Order Logic	10
2.2 Second-Order Logic	12
2.2.1 Syntax	12
2.2.2 Semantics	12
2.2.3 Inductive Definitions and Second-Order Logic	13
2.2.4 The Incompleteness of Second-Order Logic	15
2.3 Exercises	15
2.4 Bibliographic Remarks	18
3 Introduction to the Situation Calculus	19
3.1 The Situation Calculus	19
3.1.1 Intuitive Ontology for the Situation Calculus	19
3.1.2 Axiomatizing Actions in the Situation Calculus	20
3.1.3 The Qualification Problem for Actions	20
3.1.4 The Frame Problem	22
3.2 A Simple Solution to the Frame Problem (Sometimes)	23
3.2.1 Frame Axioms: Pednault's Proposal	24
3.2.2 Frame Axioms: The Davis/Haas/Schubert Proposal	26

3.2.3	A Simple Solution (Sometimes)	28
3.2.4	Aside: Normal Forms for Effect Axioms	29
3.2.5	A Simple Solution: The General Case	30
3.2.6	A Simple Solution: Functional Fluents	32
3.2.7	A Simple Solution: Summary	34
3.2.8	Some Limitations of These Action Descriptions	35
3.3	Deductive Planning with the Situation Calculus	35
3.4	Formalizing Database Transactions in the Situation Calculus	39
3.4.1	Motivation and Background	39
3.4.2	Database Updates: A Proposal	39
3.4.3	The Basic Approach: An Example	39
3.4.4	Querying a Situation Calculus Database	41
3.5	Exercises	41
3.6	Bibliographic Remarks	44
4	Foundations of the Situation Calculus	47
4.1	The Language of the Situation Calculus	47
4.2	Axioms for the Situation Calculus	48
4.2.1	Number Theory	48
4.2.2	Foundational Axioms for Situations	49
4.2.3	Some Consequences of the Foundational Axioms	52
4.2.4	Executable Situations	52
4.2.5	Further Consequences of the Foundational Axioms	53
4.3	Reasoning about Situations Using Induction	54
4.3.1	Some Examples of Inductive Proofs	55
4.3.2	State Constraints	57
4.4	Basic Theories of Action	58
4.5	Regression	61

4.6	Using Regression	67
4.6.1	Executable Ground Action Sequences	67
4.6.2	The Projection Problem and Query Evaluation	69
4.7	Regression with Functional Fluents	70
4.8	Database Logs and Historical Queries	73
4.8.1	Querying All Past Situations	74
4.8.2	Querying Some Past Situation	76
4.8.3	The Projection Problem Revisited	77
4.9	Exercises	78
4.10	Bibliographic Remarks	83
5	Implementing Basic Action Theories	85
5.1	Logical Foundations of Prolog	85
5.1.1	Why Insist on a Proper Prolog Interpreter?	87
5.1.2	More on the Equational Theory of Clark's Theorem	88
5.2	Lloyd-Topor Normal Forms for Arbitrary Definitions and Goals	90
5.2.1	What Are the Lloyd-Topor Auxiliary Predicates?	91
5.2.2	Accommodating Arbitrary Goals	92
5.2.3	Definitional Theories: Soundness, Completeness, and Closed Worlds	94
5.3	Basic Action Theories, Definitions, and Regressable Queries	95
5.3.1	Definitional Form for Action Precondition Axioms	96
5.3.2	Definitional Form for Successor State Axioms	96
5.3.3	Unfolding the Lloyd-Topor Auxiliary Predicates	102
5.3.4	Revised Lloyd-Topor Transformations	102
5.4	Exercises	107
5.5	Bibliographic Remarks	108
6	Complex Actions, Procedures, and Golog	111
6.1	Complex Actions and Procedures in the Situation Calculus	111

6.1.1	Procedures	114
6.1.2	Programs and Executable Situations	116
6.1.3	Why Macros?	117
6.1.4	Programs as Macros: What Price Must Be Paid?	118
6.1.5	Golog	119
6.2	An Elevator Controller in Golog	119
6.3	Implementation	122
6.3.1	An Interpreter	122
6.3.2	Assumptions Underlying the Implementation	125
6.3.3	Correctness of the Interpreter for Basic Action Theories with Closed Initial Database	126
6.3.4	The Elevator Example	129
6.3.5	The University of Toronto Implementation of Golog	132
6.4	Discussion	132
6.5	Proving Properties of Golog Programs	134
6.5.1	Induction Principle for While Loops	135
6.5.2	Example: A Blocks World	136
6.6	Summary	138
6.7	Exercises	140
6.8	Bibliographic Remarks	145
7	Time, Concurrency, and Processes	149
7.1	Concurrency and Instantaneous Actions	149
7.2	Concurrency via Interleaving	150
7.2.1	Examples of Interleaved Concurrency	151
7.2.2	Limitations of Interleaved Concurrency	152
7.3	The Sequential, Temporal Situation Calculus	152
7.3.1	Concurrent Temporal Processes	154
7.4	Sequential, Temporal Golog	155

7.4.1	Example: A Coffee Delivery Robot	156
7.4.2	A Singing Robot	162
7.4.3	Plan-Execution Monitoring	163
7.5	The Concurrent, Non-Temporal Situation Calculus	164
7.6	Axiomatizing Concurrent Worlds	166
7.6.1	Successor State Axioms	166
7.6.2	Action Precondition Axioms	166
7.7	The Concurrent, Temporal Situation Calculus	167
7.8	Concurrent, Temporal Golog	169
7.9	Natural Actions	170
7.9.1	Representing Physical Laws	170
7.9.2	Permissiveness of the Situation Calculus	171
7.9.3	Natural Actions and Executable Situations	172
7.9.4	An Example: Enabling Actions	173
7.9.5	Zeno's Paradox	173
7.9.6	The Natural-World Assumption	174
7.9.7	Least-Natural-Time Points	174
7.9.8	Simulating Natural Worlds	176
7.9.9	Animating Natural Worlds	180
7.10	Exercises	182
7.11	Bibliographic Remarks	183
8	Exogenous Actions, Interrupts, and Reactive Golog	185
8.1	Interrupts	185
8.2	The Semantics of RGolog	187
8.2.1	Intuitive Semantics of RGolog	187
8.2.2	Formal Semantics of RGolog	188
8.3	An RGolog Interpreter	190

8.4	Example: A Reactive Elevator Controller	191
8.4.1	A Reactive Elevator with Interactively Generated Exogenous Actions	193
8.4.2	A Reactive Elevator with Randomly Generated Exogenous Actions	197
8.5	Interrupts with Priorities	198
8.6	Discussion	199
8.6.1	Sensing and Exogenous Actions	200
8.6.2	On-Line vs. Off-Line Program Execution	200
8.7	Exercises	202
8.8	Bibliographic Remarks	203
9	Progression	205
9.1	Logical Foundations of Progression	207
9.1.1	Finite Progression Is Second-Order Definable	209
9.1.2	Progression Is Not Always First-Order Definable	210
9.1.3	But Why Not Do the Obvious?	210
9.2	Two First-Order Progressable Cases	210
9.2.1	Progressing Relatively Complete Databases	210
9.2.2	Context-Free Successor State Axioms and the Progression of Isolated Fluents	213
9.3	STRIPS Planning Systems	216
9.3.1	STRIPS Databases	216
9.3.2	STRIPS Operator Descriptions	218
9.3.3	Planning with STRIPS	221
9.4	Strongly Context-Free Successor State Axioms	223
9.5	Progression and Relational STRIPS	224
9.6	An Open-World STRIPS	226
9.7	Correctness of Relational and Open-World STRIPS	228

9.8 Exercises	230
9.9 Bibliographic Remarks	231
10 Planning	233
10.1 A Simple Breadth-First Planner	233
10.2 Example: Planning in the Blocks World	235
10.2.1 A Planning Problem Example	239
10.3 Planning with Concurrent Actions	242
10.4 OCTOPUS: A Multi-Handed Blocks World Agent	247
10.5 A Simple Depth-First Planner	255
10.6 Open-World Planning	259
10.6.1 Prime Implicates and Compiling an Initial Database	259
10.6.2 A Regression-Based Theorem-Prover	266
10.6.3 Bad Situations for Open-World Planning	268
10.6.4 Axiomatizing Incomplete Initial Situations	268
10.6.5 A Blocks World with Incomplete Initial Situation	270
10.7 Planning vs. Nondeterministic Programming	275
10.8 Exercises	277
10.9 Bibliographic Remarks	280
11 Sensing and Knowledge	283
11.1 Knowledge in the Situation Calculus	283
11.1.1 Accessibility Relations and Knowledge	284
11.1.2 Alternatives to the Initial Situation	286
11.1.3 Knowing a Referent	287
11.1.4 Knowledge and Action	287
11.1.5 Knowledge Defined	288
11.1.6 Some Consequences of This Approach	289
11.1.7 A Useful Notation	290

11.1.8	Quantifiers and Knowledge	291
11.2	Knowledge and the Designer's Perspective	292
11.2.1	Logical Omniscience Revisited	293
11.3	Knowledge-Producing Actions	293
11.4	The Frame Problem for Knowledge-Producing Actions	294
11.4.1	The No-Side-Effects Assumption for Knowledge-Producing Actions	294
11.4.2	A Successor State Axiom for K	295
11.4.3	More General Knowledge-Producing Actions	299
11.4.4	Some Consequences of this Solution	299
11.5	Accessibility in the Initial Situation	302
11.5.1	New Foundational Axioms for Situations	302
11.5.2	Some Possible Accessibility Relations	304
11.5.3	Basic Action Theories for Knowledge	305
11.6	Regression for Knowledge-Producing Actions	306
11.7	Knowledge-Based Programming	309
11.7.1	Two Simplifying Assumptions	312
11.7.2	Sense Actions	312
11.7.3	Reducing Knowledge to Provability for the Initial Situation	313
11.7.4	On-Line Execution of Knowledge-Based Programs	314
11.7.5	Reduction of Knowledge to Provability for On-Line Programs	315
11.7.6	The Dynamic Closed-World Assumption	318
11.7.7	Interpreter for Knowledge-Based Programs with Sensing	321
11.7.8	Computing Closed-World Knowledge	322
11.7.9	Putting It All Together	322
11.8	Discussion	327
11.9	Exercises	328

11.10 Bibliographic Remarks	332
12 Probability and Decision Theory	335
12.1 Stochastic Actions and Probability	335
12.1.1 How to Specify a Probabilistic Domain: A Guide for the Perplexed	339
12.1.2 Some Properties of the Specification	341
12.2 Derived Probabilities	341
12.2.1 stGolog: Stochastic Golog	345
12.3 Exogenous Events	349
12.4 Uncertainty in the Initial Situation	359
12.5 Markov Decision Processes	363
12.5.1 Sensing and stGolog Programs	366
12.5.2 Fully Observable MDPs	369
12.5.3 Partially Observable MDPs	372
12.5.4 Implementing Policies: Run-Time Sense Actions	376
12.5.5 Exogenous Actions	378
12.5.6 Solving MDP Planning Problems	378
12.6 Discussion	379
12.7 Exercises	380
12.8 Bibliographic Remarks	383
13 Concluding Remarks	385
Appendix A:	
Some Useful First-Order Inference Rules	389
A.1 Decomposition Rules	389
A.2 Generation Rules	391
A.3 Success Rules	392
A.4 Proof Rules for Unique Names Axioms	392
A.5 Simplification Rules	393

A.6	Additional Success Rules	394
A.7	Examples of Proofs	394
A.8	Exercises	399
Appendix B:		
	The Qualification and Ramification Problems	401
B.1	Constraints and the Ramification Problem	401
B.2	Constraints and the Qualification Problem	402
B.3	Solving the Qualification and Ramification Problems	404
Appendix C:		
	Resources	407
	References	409
	Index	419