

CONTENTS

I Introduction	1
1. INTRODUCTION.	2
1.1. Inference and Learning with Planning Models	2
1.2. Overview of contributions	5
1.3. A Note on Software	6
1.4. Declaration of Previous Work.	6
1.5. Thesis Outline	7
II Background	9
2. CLASSICAL PLANNING	10
2.1. The Classical Planning Problem	10
2.2. The Planning Domain Definition Language	12
2.2.1. The STRIPS Fragment of PDDL	12
2.2.2. Conditional Effects.	15
2.3. Complexity of STRIPS Planning	16
2.4. Main Approaches to Classical Planning	17
2.4.1. Classical Planning as Heuristic Search.	17
2.4.2. Classical Planning as Propositional Satisfiability	18
2.5. Planning with Temporally Extended Goals	19
2.6. Planning with Sensing	20
III The Learning and Inference Framework	22
3. FRAMEWORK	23
3.1. The Observer	23
3.1.1. The Sensor Model	23
3.1.2. Deterministic and Non-Deterministic Sensor Models.	27
3.1.3. Extending a Sensor Model with Costs	28

3.2. A Theory for Acting and Sensing.	30
3.2.1. Fundamental Assumptions	30
3.2.2. The Scope of the Acting-Sensing Integration	32

IV Inference 35

4. OBSERVATION DECODING	36
4.1. Working Example: Blindspots	36
4.2. The Observation Decoding Problem	39
4.3. The Solution of the Observation Decoding Problem	43
4.4. Observation Decoding as Planning	44
4.4.1. LTL Preliminaries	45
4.4.2. A Monitor Automaton for the Observation Sequence.	46
4.4.3. The Compilation	47
4.4.4. Properties of the compilation	53
4.4.5. Computing the most likely explanation	54
4.4.6. A walk-through of the compilation.	55
4.5. Experimental evaluation.	58
4.5.1. Evaluation Benchmark.	59
4.5.2. Results	60
4.6. Discussion	61
5. TEMPORAL INFERENCE	63
5.1. Working Example: Driverlog	63
5.2. Hypotheses	67
5.3. The Temporal Inference problem	69
5.4. Special Classes of Temporal Inference Problems	71
5.5. The Solution to the Temporal Inference Problem	74
5.6. Temporal Inference as Planning.	75
5.7. Experimental evaluation.	76
5.7.1. Empirical Results	77
5.8. Discussion	78

6. MODEL RECOGNITION	80
6.1. Working Example: Navigation Strategies	80
6.2. The Model Recognition Problem	82
6.3. The solution to the Model Recognition Problem	86
6.4. Recognition of STRIPS Action Models	87
6.4.1. Edit costs for STRIPS schematic action models	90
6.5. Model Recognition as planning	91
6.5.1. The Alphabet of a STRIPS Schematic Action Model	93
6.5.2. The Compilation	97
6.6. Experimental Evaluation	103
6.6.1. Use Case 1: Recognition of <i>regular automata</i>	103
6.6.2. Use Case 2: Recognizing Failures in a Non-deterministic <i>Blocksworld</i>	104
6.6.3. Use Case 3: Recognition of Navigation Strategies	106
6.7. Discussion	106

V Learning 108

7. ACTION MODEL LEARNING	109
7.1. On the Use of Planning for Action Model Learning	109
7.2. The Action Model Learning Problem	113
7.3. The Solution to the Action Model Learning Problem.	114
7.4. Action Model Learning as Planning	115
7.5. Evaluation of action models.	118
7.5.1. Syntactic-based precision and recall for action models	119
7.5.2. Semantic-based precision and recall for action models	120
7.6. Experimental Evaluation	123
7.6.1. Setup	123
7.6.2. Impact of the size of the set of learning examples.	124
7.6.3. Comparison with ARMS	128
7.6.4. Learning with minimal input knowledge.	129
7.6.5. Syntactic versus semantic evaluation.	132
7.7. Discussion	134

VI Conclusions and Future Work	137
8. CONCLUSIONS	138
8.1. Summary of Contributions	138
8.2. Future work.	139
BIBLIOGRAPHY.	142