

Contents

	<i>page</i>
<i>Preface</i>	<i>ix</i>
1 Declarative programming in AnsProlog [*] : introduction and preliminaries	1
1.1 Motivation: Why AnsProlog [*] ?	3
1.2 Answer set frameworks and programs	8
1.3 Semantics of AnsProlog [*] programs	16
1.4 Database queries and AnsProlog [*] functions	40
1.5 Notes and references	44
2 Simple modules for declarative programming with answer sets	46
2.1 Declarative problem solving modules	47
2.2 Knowledge representation and reasoning modules	73
2.3 Notes and references	81
3 Principles and properties of declarative programming with answer sets	83
3.1 Basic notions and basic properties	84
3.2 Some AnsProlog [*] sub-classes and their basic properties	93
3.3 Restricted monotonicity and signed AnsProlog [*] programs	108
3.4 Analyzing AnsProlog [*] programs using ‘splitting’	113
3.5 Language independence and language tolerance	120
3.6 Interpolating an AnsProlog program	126
3.7 Building and refining programs from components: functional specifications and realization theorems	137
3.8 Filter-abducible AnsProlog ^{¬,OR} programs	144
3.9 Equivalence of programs and semantics preserving transformations	154
3.10 Notes and references	168
4 Declarative problem solving and reasoning in AnsProlog [*]	170
4.1 Three well-known problem solving tasks	170
4.2 Constraint satisfaction problems (CSPs)	183

vi	<i>Contents</i>	
4.3	Dynamic constraint satisfaction problems (DCSPs)	186
4.4	Combinatorial graph problems	188
4.5	Prioritized defaults and inheritance hierarchies	192
4.6	Notes and references	197
5	Reasoning about actions and planning in AnsProlog*	199
5.1	Reasoning in the action description language \mathcal{A}	199
5.2	Reasoning about actions and plan verification in richer domains	229
5.3	Answer set planning examples in extensions of \mathcal{A} and STRIPS	244
5.4	Approximate planning when initial state is incomplete	261
5.5	Planning with procedural constraints	262
5.6	Explaining observations through action occurrences and application to diagnosis	269
5.7	Case study: Planning and plan correctness in a space shuttle reaction control system	274
5.8	Notes and references	277
6	Complexity, expressiveness, and other properties of AnsProlog* programs	278
6.1	Complexity and expressiveness	278
6.2	Complexity of AnsDatalog* sub-classes	288
6.3	Expressiveness of AnsDatalog* sub-classes	301
6.4	Complexity and expressiveness of AnsProlog* sub-classes	304
6.5	Compact representation and compilability of AnsProlog	311
6.6	Relationship with other knowledge representation formalisms	313
6.7	Notes and references	341
7	Answer set computing algorithms	345
7.1	Branch and bound with WFS: wfs-bb	346
7.2	The assume-and-reduce algorithm of SLG	358
7.3	The smodels algorithm	363
7.4	The dlv algorithm	372
7.5	Notes and references	379
8	Query answering and answer set computing systems	382
8.1	Smodels	382
8.2	The dlv system	403
8.3	Applications of answer set computing systems	412
8.4	Pure PROLOG	440
8.5	Notes and references	456
9	Further extensions of and alternatives to AnsProlog*	458
9.1	AnsProlog ^{not, or, ¬, ⊥} : allowing not in the head	458
9.2	AnsProlog ^{(not, or, ¬, ⊥)*} : allowing nested expressions	461
9.3	AnsProlog ^{¬, or, K, M} : allowing knowledge and belief operators	466

<i>Contents</i>		vii
9.4	Abductive reasoning with AnsProlog: AnsProlog ^{abd}	470
9.5	Domain closure and the universal query problem	471
9.6	AnsProlog _{set} : adding set constructs to AnsProlog	475
9.7	AnsProlog- ^{<} programs: AnsProlog ^{<} programs with ordering	477
9.8	Well-founded semantics of programs with AnsProlog syntax	482
9.9	Well-founded semantics of programs with AnsProlog ^{<} syntax	490
9.10	Notes and references	492
Appendix A: Ordinals, lattices, and fixpoint theory		494
Appendix B: Turing machines		496
<i>Bibliography</i>		498
<i>Index of notation</i>		519
<i>Index of terms</i>		522