

Table of Contents

PART I – IMPLEMENTING NEGATION

1 IMPLEMENTING NEGATION AS FAILURE	1
1.1 Introduction	1
1.2 Unsound Implementations	3
1.2.1 Not and Not Equals	3
1.2.2 If Then Else	3
1.2.3 Soft Cut and Cases	4
1.2.4 All Solutions Predicates	4
1.2.5 Epilog and Two Level PROLOG	4
1.3 Delaying Until Ground	4
1.3.1 Universal Quantifiers	5
1.4 IC-PROLOG Not	5
1.4.1 Retrying Instead of Aborting	6
1.4.1.1 Minimizing the Number of Retries	6
1.4.1.2 Infinite Numbers of Solutions	7
1.5 Inequality	8
1.6 Incremental Not	9
1.7 All Solutions Predicates	10
1.7.1 Introduction	10
1.7.2 Background	11
1.7.2.1 Specification	11
1.7.2.2 An Example	11
1.7.2.3 Implementations	12
1.7.3 Problems with Previous Work	13
1.7.3.1 Declarative Semantics	13
1.7.3.2 Negation	13
1.7.3.3 Lists are Not Sets	14
1.7.3.4 Variables in Answers	14
1.7.4 A Sound Implementation	15
1.7.4.1 Representation of Sets	15
1.7.4.2 Removing Duplicates and Sorting	16
1.7.4.3 Negation	16
1.7.5 Application to Negation	18
1.7.6 Summary	18
1.8 Negative Unifiers	19
1.9 Program Transformation	19
1.9.1 Functions	21
1.10 Conclusions	24

PART II – CONTROL OF LOGIC PROGRAMS

2 INTRODUCTION	25
3 AUTOMATING CONTROL	26
3.1 Introduction	26
3.2 The Execution Model	26
3.3 System Predicates	27
3.4 Database Procedures	28
3.4.1 Priority Declarations	29
3.4.2 Minimizing the Number of Unifications	30
3.5 Recursive Procedures	30
3.5.1 Wait Declarations	31
3.5.2 Generating Wait Declarations	32
3.5.3 The Algorithm	34
3.5.4 Difficulties With the Algorithm	34
3.5.5 Ordering Subgoals	36
3.6 A Program to Generate Control Information	37
3.7 Further Examples	40
3.7.1 Systems Programming	40
3.7.2 Object Oriented Programming	42
3.7.3 Infinite Streams	44
3.8 Related Work	45
3.8.1 PROLOG M	45
3.8.2 Data Flow Analysis	45
3.8.3 Capture Rules	46
3.9 Other Control Primitives	46
3.9.1 Local Control	47
3.9.2 Non-local Control	48
3.10 Summary	49
4 HETEROGENEOUS SLD RESOLUTION	50
4.1 Introduction	50
4.2 Definitions	50
4.2.1 SLD Resolution	50
4.2.2 HSLD Resolution	51
4.3 Theoretical Results	52
4.4 An Application	54
4.5 Summary	55
5 AN OVERVIEW OF PROLOG CONTROL RULES	57
5.1 Introduction	57
5.2 Program Examples	57
5.3 Desirable Properties of Computation Rules	59
5.3.1 Minimizing the Search Tree Size	59
5.3.2 Detecting Failure	59
5.3.3 Avoiding Failure	59
5.4 Control Rules	59

5.4.1 Minimizing the Search Tree Size	60
5.4.1.1 Select Calls Which Fail	60
5.4.1.2 Select Locally Deterministic Calls	60
5.4.1.3 Minimizing the Number of Matching Clauses	61
5.4.1.4 Database Queries	62
5.4.2 Detecting Failure	62
5.4.2.1 Call Tests as Soon as Possible	62
5.4.2.2 Resume Woken Calls as Soon as Possible	62
5.4.2.3 Eager Consumers	62
5.4.2.4 The Killer Heuristic	63
5.4.2.5 Fairness	63
5.4.2.6 Breadth First	63
5.4.2.7 Pseudo Parallelism	64
5.4.2.8 Avoid Left Recursion	64
5.4.3 Avoiding Failure	64
5.4.3.1 Freeze	64
5.4.3.2 Lazy Producers	65
5.4.3.3 Wait Declarations	65
5.4.3.4 Triggers and When Declarations	66
5.4.3.5 Reordering Clause Bodies	66
5.4.3.6 Delaying System Predicates	67
5.5 Discussion	67
5.6 Conclusions	68
6 CONCLUSION	69
REFERENCES	70
APPENDIX 1 – THE MU-PROLOG REFERENCE MANUAL	77
APPENDIX 2 – THE MU-PROLOG DEDUCTIVE DATABASE	101