

Contents

Abstract	iii
Acknowledgements	iii
1 Introduction	1
1.1 Lazy Evaluation	2
1.2 A Logic Programming Scheme	4
1.3 Semantics	5
1.4 Logic+Functional Programming	6
1.5 Constraint Logic Programming	7
1.6 Control Generation	7
2 Basic Notions	9
2.1 First-Order Logic	9
2.1.1 Syntax	9
2.1.2 Semantics	11
2.2 Logic Programming	12
2.2.1 Syntax	12
2.2.2 Declarative Semantics	14
2.2.3 SLD-Resolution and Proof Trees	16
2.2.4 SLDNF-Resolution	18
2.3 Further Notations	19
2.3.1 Set Theory	19
2.3.2 Occurrences	19
3 A Logic Programming Scheme	21
3.1 Introduction	21
3.1.1 Our Approach — Basic Ideas	22
3.1.2 An Example: Equation Solving	23
3.2 Inference Rules and Selection Rules	24
3.2.1 Inference Rules	24
3.2.2 Soundness of Inference rules	26
3.2.3 Selection Rules	28
3.3 The LLP Scheme	29
3.3.1 Definitions	29
3.3.2 Defining SLDNF-Resolution in the LLP scheme	32

3.3.3	Some Remarks on the LLP scheme	36
3.4	Properties of the LLP Scheme	37
3.4.1	Soundness	37
3.4.2	(In)Completeness	38
3.4.3	Splitting Programs	40
3.5	Examples and Programming Techniques	41
3.5.1	Sepia's Delay clauses	42
3.5.2	Fair Parallel Selection Rule	42
3.5.3	SLS-Resolution	43
3.6	Related Work	43
3.6.1	Logic Programming Schemes	43
3.6.2	Multiple Inference Rules in Logic Programming	44
4	Control Generation for Logic Programs	47
4.1	Introduction	47
4.1.1	NU-Prolog	48
4.1.2	Motivating Examples	50
4.2	A Language for Control	52
4.2.1	How to Specify Computation Rules	52
4.2.2	What Should When-Declarations Look Like?	53
4.3	An Algebra of When-Declarations	56
4.3.1	When-Declarations	56
4.3.2	A Partial Order on When-Declarations	58
4.3.3	Safe When-Declarations and Programs	61
4.3.4	Good Computation Rules	66
4.4	Algorithms for Generating When-Declarations	66
4.4.1	The Basic Algorithm Scheme	67
4.4.2	Deriving Deterministic Algorithms from the Scheme	69
4.4.3	Practical Results	73
4.4.4	Generalization of When-Declarations	86
4.5	Extensions	88
4.5.1	Most Specific Programs	88
4.5.2	Normal Programs	91
4.6	Discussion	92
4.6.1	Summary of Our Approach	92
4.6.2	Applications	92
4.6.3	Related work	92
5	Logic+Functional Programming	97
5.1	Introduction	97
5.2	Syntax	99
5.2.1	Informal Operational Semantics	101
5.3	Compiling Logic+Functional Programs	101
5.3.1	Removing the Overlapping of Equations	102
5.3.2	Flattening Function Calls	103
5.3.3	Relational Notation for Functions	106
5.3.4	Properties of the Transformed Program	107

5.4	Declarative Semantics	108
5.4.1	Introduction	108
5.4.2	Extending the Well-Founded Semantics	108
5.4.3	Functions in the Well-Founded Semantics	111
5.5	Operational Semantics	112
5.5.1	SLDNF-Resolution	112
5.5.2	Partial Evaluation and Simplification	112
5.5.3	Backward Evaluation	113
5.6	Lazy Evaluation	115
5.6.1	Introduction	115
5.6.2	Extending the Declarative Semantics	116
5.6.3	Properties of the Extended Declarative Semantics	116
5.6.4	Operational Semantics	118
5.7	Implementing full Miranda	124
5.7.1	Implementing Higher-Order Functions	124
5.7.2	Eliminating Local Definitions	128
5.7.3	'Fully Lazy' Evaluation	130
5.8	Discussion	131
6	Constraint Logic Programming	133
6.1	Introduction	133
6.2	CLP(X)	133
6.2.1	Introduction to CLP(X)	134
6.2.2	LLP Procedures for CLP(X)	135
6.2.3	Final Remarks	142
6.3	Consistency Techniques with Finite Domains	142
6.3.1	An Introduction to Van Hentenryck's Approach	143
6.3.2	Basic Simulation	143
6.3.3	Extending the Domain Concept	146
6.3.4	More Inference Rules	148
6.3.5	Discussion	149
6.4	Conclusions	151
7	Conclusions	153
	Bibliography	155
	Lebenslauf	163