

Harold Boley

A Tight, Practical Integration of Relations and Functions



Springer

Contents

1	An Overview of the Relational-Functional Language RELFUN	1
1.1	Declarative Merger via Minimal Extensions	1
1.2	From Relations and Functions to Operators	5
1.3	PROLOG-LISP-RELFUN Comparison	8
1.4	Semantics and Implementation	11
1.5	Applications	14
1.6	Related Work	16
1.7	Reader's Guide	19
2	Extended Logic-plus-Functional Programming	21
2.1	Introduction	21
2.2	Relations Defined by Hornish Clauses	23
2.2.1	Open-world DATALOG	23
2.2.2	PROLOG-like Structures and Lists	25
2.2.3	Varying-arity Structures	26
2.2.4	Varying-arity Relationships	28
2.2.5	Higher-order Constructors and Relations	29
2.3	Functions Defined by Footed Clauses	31
2.3.1	DATAFUN as a Functional Database Language	31
2.3.1.1	Molecular Rules and Non-ground Functions	31
2.3.1.2	Footed Rules and the density Example	33

2.3.1.3	Non-determinism, DATALOG Relationalizing, and WAM Compilation	35
2.3.2	Full RELFUN Exemplified by “Self”-Functions	36
2.3.3	Higher-order Constructors and Functions	39
2.4	The Logic/Functional Style in Use	42
2.4.1	<code>serialise</code> : Inplace Specialization of Structures	42
2.4.2	<code>wang</code> : On-the-fly Construction of Proof Trees	45
2.4.3	<code>eval</code> : Interpreting a LISP Subset in RELFUN	48
2.5	Conclusions	50
2.6	Appendix: The RELFUN Syntax	52
3	A Direct Semantic Characterization of RELFUN	55
3.1	Introduction	55
3.2	Extending First-order Theories to First-order Relational-Functional Theories	61
3.3	Relational-Functional Interpretations and Models	65
3.4	SLV-Resolution	72
3.5	Soundness of SLV-Resolution	78
3.6	Least Herbrand Crossbase Models as Fixpoints	80
3.7	Completeness of SLV-Resolution	85
3.8	Conclusions	87
4	Finite Domains and Exclusions as First-class Citizens	89
4.1	Introduction	89
4.2	Domain Terms	91
4.3	Exclusion Terms	93
4.4	Occurrence Bindings	95
4.5	Domains/Exclusions in Relation Definitions	97
4.5.1	Facts and <code>dom/exc</code> Reductions	97
4.5.2	Clauses and <code>bnd-to-“.=”</code> Reductions	99

4.6	Finite-Domain/Exclusion Functional Programming	101
4.6.1	Domains/Exclusions as Function Arguments	102
4.6.2	Functions with Domain/Exclusion Values	103
4.7	Domain and Exclusion Anti-unification	105
4.8	Operational Semantics	109
4.9	Conclusions	111
4.10	Appendix: The RELFUN Meta-unify	114
5	Multiple-valued Horn Clauses and Their WAM Compilation	117
5.1	Introduction	117
5.2	A Multiple-valued Relational/Functional Language	120
5.2.1	Amalgamating Relations and Functions	120
5.2.2	Single-valued and Multiple-valued Clauses	123
5.2.3	An Example: Refining the <i>palindrome</i> Operator	128
5.2.4	Higher-order Functions and Relations	134
5.3	Relational/Functional WAM Compilation	137
5.3.1	A Compilation Strategy	137
5.3.2	Evaluative Foots and Denotative Normalization	140
5.3.3	Non-deterministic, Multiple-valued Nestings and Static Flat- tening	143
5.3.4	Higher-order Clauses and Constant-operator Reduction	148
5.3.5	Translation to WAM Instructions	151
5.4	Conclusions	157
	Bibliography	161