

Contents

1	Introduction	1
1.1	Main Target Group of the Book	2
1.2	Prerequisites Needed to Understand the Book	3
1.3	Content	3
1.4	Logical Organization of the Book	4
1.5	Structure of the Chapters and Book Webpage	4
2	Semantic Web	7
2.1	Introduction	7
2.2	Overview	8
2.3	RDF Data	9
2.3.1	N3 Notation	11
2.3.2	RDF/XML	13
2.4	Ontology Languages	13
2.5	Open World Assumption	16
2.6	No Unique Name Assumption	17
2.7	SPARQL Query Language	17
2.7.1	Language Constructs of SPARQL	18
2.7.2	SPARQL Protocol for RDF	24
2.7.3	SPARQL Query Results XML Format	26
2.7.4	RDF Stores	27
2.8	Rules	28
2.9	Related Work	31
2.9.1	RIF Processing	31
2.9.2	Optimizations for Recursive Rules	33
2.10	Summary and Conclusions	34
3	External Sorting and B⁺-Trees	35
3.1	Motivation	35
3.2	B ⁺ -trees	36
3.2.1	Properties of B ⁺ -Trees	37

3.2.2 Self-balancing Property of B ⁺ -Trees	38
3.2.3 Searching	39
3.2.4 Prefix Search in Combination with Sideways Information Passing	39
3.2.5 Inserting	41
3.2.6 Deleting	43
3.2.7 B ⁺ -Tree Construction from a large Dataset	45
3.3 Heap	45
3.4 (External) Merge Sort	47
3.5 Replacement Selection	48
3.6 External Chunks Merge Sort	50
3.7 Distribution Sort	52
3.8 RDF Distribution Sort	53
3.9 Experimental Analysis	56
3.9.1 SP ² B Dataset	57
3.9.2 Yago Dataset	58
3.10 Summary and Conclusions	63
4 Query Processing Overview	67
4.1 The LUPOSDATE System	67
4.2 Phases of Query Processing	69
4.3 CoreSPARQL	73
4.3.1 Defining CoreSPARQL	73
4.3.2 Transforming SPARQL Queries into CoreSPARQL Queries	74
4.3.3 CoreSPARQL Grammar	77
4.4 Related Work	78
4.5 Summary and Conclusions	78
5 Logical Optimization	79
5.1 Logical Algebra	79
5.1.1 Semantics of the Logical Algebra Operators	81
5.2 Logical Optimization Rules	85
5.2.1 Pushing FILTER Operators	85
5.2.2 Splitting and Commutativity of FILTER Operators	87
5.2.3 Constant and Variable Propagation	87
5.2.4 Heuristic Query Optimization Using Equivalency Rules	89
5.2.5 Cost-Based Optimization	90
5.2.6 Histograms	99
5.3 Further Related Work	101
5.4 Summary and Conclusions	101
6 Physical Optimization	103
6.1 Motivation	104
6.2 Related Work	106

6.3	Indexing	108
6.3.1	Building In-Memory Indices	109
6.3.2	Building Disk-Based Indices	110
6.4	Pipelining Versus Materialization	116
6.4.1	Pipeline-Breaker	116
6.4.2	Sideways Information Passing	116
6.5	Join Algorithms	117
6.5.1	Nested-Loop Join	117
6.5.2	Merge Join	120
6.5.3	Index Join	122
6.5.4	Hash Join	123
6.6	Dynamically Restricting Triple Patterns	126
6.7	Sorting Numbering Scheme	129
6.7.1	Joins Without Presorting Numbers	129
6.7.2	Joins with Presorting Numbers	131
6.7.3	Optimization of Fast Sorting	132
6.7.4	Sorting for Complex Joins	132
6.7.5	Additional Benefits from SIP Strategies	135
6.8	Optional	136
6.8.1	MergeOptional	136
6.9	Duplicate Elimination	137
6.9.1	Duplicate Elimination Using Hashing	137
6.9.2	Duplicate Elimination Using Sorting	138
6.9.3	Duplicate Elimination Using Presorting Numbers	138
6.10	Cost Model	138
6.11	Performance Evaluation	139
6.11.1	Performance Evaluation for In-memory Databases	139
6.11.2	Performance Evaluation for Large-Scale Datasets	145
6.12	Summary and Conclusions	152
7	Streams	155
7.1	Introduction	155
7.2	eBay	156
7.3	Monitoring eBay Auctions	157
7.3.1	Monitoring System	157
7.3.2	Demonstration	158
7.3.3	Streaming SPARQL Engine	159
7.4	Special Operators for Stream Processing	160
7.4.1	Types of Stream Operators	160
7.4.2	Types of Window Operators	161
7.5	Related Work	161
7.5.1	Data Streams in General	161
7.5.2	Semantic Web Data Streams	162
7.6	Summary and Conclusions	162

- 8 Parallel Databases** 163
 - 8.1 Motivation 163
 - 8.2 Types of Parallelisms 165
 - 8.3 Amdahl’s Law 167
 - 8.4 Parallel Monitors and Bounded Buffers 168
 - 8.5 Parallel Join Using a Distribution Thread 168
 - 8.6 Parallel Merge Join Using Partitioned Input 169
 - 8.7 Parallel Computation of Operands 172
 - 8.8 Performance Evaluation 173
 - 8.9 Performance Gains and Loss 175
 - 8.10 Summary and Conclusions 175
- 9 Inference** 177
 - 9.1 Introduction 177
 - 9.2 RDF Schema Inference Rules 178
 - 9.3 Materialization of Inference and Consequences
for Query Optimization 179
 - 9.4 Logical Optimization for Inference 180
 - 9.5 Performance Analysis 187
 - 9.6 Related Work 189
 - 9.7 Summary and Conclusions 189
- 10 Visual Query Languages** 191
 - 10.1 Motivation 191
 - 10.2 Related Work 193
 - 10.3 RDF Visual Editor 194
 - 10.4 SPARQL Visual Editor 194
 - 10.5 Browser-Like Query Creation 194
 - 10.6 Generating Condensed Data View 196
 - 10.7 Refining Queries 197
 - 10.8 Query Formulation Demo 198
 - 10.9 Computation of Suggested Triple Patterns for Query
Refinement 199
 - 10.10 Summary and Conclusions 201
- 11 Embedded Languages** 203
 - 11.1 Motivation 203
 - 11.2 Related Work 204
 - 11.3 Embedding Semantic Web Languages Into JAVA 205
 - 11.3.1 The Type System 208
 - 11.3.2 Subtype Test 210
 - 11.3.3 Satisfiability Test of Embedded SPARQL
and SPARUL Queries 215
 - 11.3.4 Determination of the Query Result Types 217
 - 11.4 Summary and Conclusions 217

- 12 Comparison of the XML and Semantic Web Worlds** 219
 - 12.1 Introduction 219
 - 12.2 Concepts and Visions 221
 - 12.3 Data Models 221
 - 12.4 Schema and Ontology Languages 222
 - 12.5 Query Languages 223
 - 12.6 Embedding SPARQL into XQuery/XSLT 226
 - 12.6.1 Embedded SPARQL 226
 - 12.6.2 Translation Process 229
 - 12.6.3 Experimental Analysis 235
 - 12.7 Embedding XPath Into SPARQL 240
 - 12.7.1 Translation of XPath Subqueries Into SPARQL
 Queries 241
 - 12.7.2 Performance Analysis 247
 - 12.8 Related Work 248
 - 12.9 Summary and Conclusions 250

- 13 Summary, Conclusions, and Future Work** 251
 - 13.1 Possibilities for Future Work 252

- References** 255

- Index** 267