

Contents

1	Introduction	1
1.1	Fundamentals of Term Indexing	4
1.2	Original Contributions of this Book	12
1.3	Reader's Guide	14
2	Preliminaries	17
2.1	Rules	17
2.2	First-Order Logic	18
2.3	Graphs and Trees	24
2.4	Algorithms	25
3	Fundamental Data Structures and Algorithms	27
3.1	Memory Management	28
3.2	Terms	29
3.3	Substitutions, Matching, and Unification	35
4	Attribute-Based Indexing	43
4.1	Matching Pretest	44
4.2	Outline Indexing	45
4.3	Superimposed Codewords	47
5	Set-Based Indexing	51
5.1	Top Symbol Hashing	52
5.2	Coordinate Indexing	53
5.3	Path Indexing	56
5.3.1	Standard Path Indexing	58
5.3.2	Dynamic Path Indexing	87
5.3.3	Extended Path Indexing	90
5.3.4	Path Indexing for Subterm Retrieval	113
5.3.5	Path Indexing for AC-Theories	115

6	Tree-Based Indexing	127
6.1	Discrimination Tree Indexing	127
6.1.1	Standard Discrimination Trees	128
6.1.2	Perfect Discrimination Trees	138
6.1.3	Deterministic Discrimination Trees	142
6.1.4	Adaptive Discrimination Trees	146
6.2	Abstraction Tree Indexing	150
6.2.1	Standard Abstraction Trees	150
6.2.2	Linear Abstraction Trees	157
6.3	Substitution Tree Indexing	158
6.3.1	Standard Substitution Trees	160
6.3.2	Weighted Substitution Trees	189
6.3.3	Linear Substitution Trees	191
6.3.4	Compiled Substitution Trees	193
7	Comparison of Indexing Techniques	201
7.1	Data Structures	202
7.2	Soundness and Completeness	204
7.3	Functionality	205
7.4	Complexity	210
7.5	Performance	212
8	Indexing in Theorem Provers	233
8.1	Binary Resolution	234
8.2	Binary Resolution with Theory Unification	234
8.3	Hyperresolution	235
8.4	Unit Resulting Resolution	236
8.4.1	PURR	237
8.5	Subsumption	248
8.6	Paramodulation	256
8.7	Demodulation	257
8.8	Completion	259
9	Summary	265
9.1	A Short Survey on the Presented Indexing Techniques	266
9.2	Main Results	268
9.3	Conclusions	270
9.4	The Future	272
	Bibliography	275
	Index	281

List of Figures

1. Introduction	1
1.1 Retrieval of type $n : 1$	8
1.2 Retrieval of type $n : m$	9
1.3 Insertion and deletion of type $n : 1$	10
1.4 Insertion of type $n : m$	10
2. Preliminaries	17
2.1 Presentation of algorithms	25
3. Fundamental Data Structures and Algorithms	27
3.1 Simple and efficient memory management	28
3.2 Conventional, flatterm, and PROLOG term structure of $f(x, f(a, x))$	30
3.3 Algorithm for traversing a conventional term	30
3.4 Algorithm for traversing a flatterm	31
3.5 Algorithm for traversing a PROLOG term	32
3.6 A conventional term and its shared version	34
3.7 Algorithm for integrating a term into a shared structure	34
3.8 Algorithm for the matching of terms	37
3.9 Algorithm for the unification of terms	40
4. Attribute-Based Indexing	43
5. Set-Based Indexing	51
5.1 Paths of the term $t = h(a, g(b), x)$	59
5.2 Standard path index	66
5.3 Accessing the path lists using a hash table	67
5.4 Accessing the path lists using a trie	68
5.5 Minimal trie for index depth 1	69
5.6 Query tree for $\text{variants}_{SPI}(\mathcal{I}, h(a, g(b), x), 2)$	71
5.7 Query tree for $\text{instances}_{SPI}(\mathcal{I}, h(a, g(b), x), 2)$	71

5.8	Query tree for $\text{unifiable}_{SPI}(\mathcal{I}, h(a, g(b), x), 2)$	72
5.9	Query tree for $\text{general}_{SPI}(\mathcal{I}, h(a, g(b), x), 2)$	72
5.10	Standard query tree simplification rules SR	73
5.11	Minimizing the query tree for $\text{unifiable}_{SPI}(\mathcal{I}, f(a, g(b)), 2)$	75
5.12	Algorithm for inserting terms into a standard path index	76
5.13	Algorithm for sorted list retrieval	78
5.14	Algorithm for an entry-oriented search in a query tree	79
5.15	Retrieval in sorted path lists	80
5.16	Retrieval in sorted path lists (cont.)	81
5.17	Retrieval in sorted path lists (cont.)	81
5.18	Retrieval in sorted path lists (cont.)	82
5.19	Retrieval in sorted path lists (cont.)	82
5.20	Algorithm for a set-oriented search in a query tree	84
5.21	Algorithm for the marking technique	84
5.22	Evaluation of a query tree using the marking technique	85
5.23	How retrieval and insertion effect the trie	88
5.24	Query tree for $\text{unifiable}_{DPI}(\mathcal{I}, f(a, g(b)), 1)$	90
5.25	Cooperation of infirm matchability and standard path indexing	94
5.26	Cooperation of infirm unifiability and standard path indexing	95
5.27	Distances in $t = h(a, h(a, h(a, b, g(g(c))), g(g(c))), g(g(c)))$ to b . .	97
5.28	Extended path index	106
5.29	Extended query tree for $\text{unifiable}_{EPI}(\mathcal{I}, f(x, g(x)), 2, 1)$	107
5.30	Extended query tree simplification rules ER	108
5.31	Minimizing $\text{unifiable}_{EPI}(\mathcal{I}, f(g(x), g(h(x))), 3, 2)$	110
5.32	Algorithm for inserting terms into an extended path index	112
5.33	Minimal trie of depth 2 used for subterm retrieval	114
5.34	Query tree with combinator	125
5.35	Algorithm for an entry-oriented search with a combinator	126
6.	Tree-Based Indexing	127
6.1	Standard discrimination tree	129
6.2	Node structure in a standard discrimination tree	133
6.3	Algorithm for general_{SDT}	134
6.4	Algorithm for merge_{SDT}	136
6.5	Perfect discrimination tree	138
6.6	Algorithm for merge_{PDT}	141
6.7	Deterministic discrimination tree	143
6.8	Node structure in a deterministic discrimination tree	144
6.9	Algorithm for general_{DDT}	145
6.10	Adaptive discrimination tree	147

6.11	An interchange step in an adaptive discrimination tree	149
6.12	Abstraction tree	151
6.13	Algorithm for unifiable _{SAT}	155
6.14	Merging two abstraction trees	156
6.15	Substitution tree	159
6.16	Standard substitution tree	161
6.17	Algorithm for search	171
6.18	Bindings during the retrieval for the query $\{u \mapsto f(a, y)\}$	172
6.19	Insertion sequence for substitutions with identical domains	175
6.20	Insertion sequence for substitutions with different domains	176
6.21	Insertion of $f(c, g^n(a))$ into an optimal substitution tree	180
6.22	Algorithm for merge _{ST}	183
6.23	Merging of two substitution trees	184
6.24	Bindings during merge of two substitution trees	185
6.25	Subsumption as an $n : m$ indexing task	188
6.26	Subsumption in weighted substitution trees	190
6.27	Standard and linear substitution tree	191
6.28	Simple substitution tree	194
6.29	PROLOG representation of a simple substitution tree for finding unifiable terms	196
7.	Comparison of Indexing Techniques	201
7.1	Correlations between tree-based indexing techniques	203
7.2	Features of indexing techniques	206
7.3	Representation of substitutions	210
7.4	Memory requirements [KBytes]	215
7.5	Average memory requirements	216
7.6	Memory requirements of random terms	217
7.7	Insertion times [Seconds]	218
7.8	Average insertion times	219
7.9	Deletion times [Seconds]	219
7.10	Average deletion times	220
7.11	Retrieval of instances [Seconds]	221
7.12	Average retrieval of instances	221
7.13	Retrieval of instances of AVG (1000)	222
7.14	Retrieval of generalizations [Seconds]	223
7.15	Average retrieval of generalizations	224
7.16	Retrieval of generalizations of AVG (1000)	225
7.17	Retrieval of unifiable terms [Seconds]	226
7.18	Average retrieval of unifiable terms	227

7.19	Retrieval of terms unifiable with AVG (1000)	228
7.20	Merge [Seconds]	229
7.21	Average retrieval using the merge	229
7.22	Summary of the experiments	230
8.	Indexing in Theorem Provers	233
8.1	Clause graph for distributed unit resulting resolution	239
8.2	Clause process in Aßmann's concept	240
8.3	Resolution process in PURR	242
8.4	Terminator process in PURR	244
8.5	Experiments with the implicational propositional calculus	245
8.6	Experiments with the equivalential calculus	246
8.7	Experiments with the R calculus	247
8.8	Primitive forward subsumption	249
8.9	Standard forward subsumption	249
8.10	Indexed forward subsumption	250
8.11	Primitive backward subsumption	250
8.12	Standard backward subsumption	251
8.13	Indexed backward subsumption	252
8.14	Optimized forward subsumption	252
8.15	Experiments with forward subsumption	254
8.16	Experiments with backward subsumption	255
8.17	Knuth-Bendix completion	260
8.18	Experiments with completion	263
9.	Summary	265