

Subject Index

- Σ -algebra homomorphism, 374
- $\lambda\sigma$ -calculus, 359
- π -calculus, 38, 380
- Δ -GRAMMARS, 235, 236, 238, 239
- Δ -productions, 237–241, 244–246
- Δ -production, 236

- A'UM, 61, 211, 217, 219
- A* search, 182
- ABCL, 108, 131, 132
- ABCL/1, 107, 112, 125, 138, 177, 337
- ABCL/onAP1000, 131, 133
- ABCL/onEM-4, 107
- ABCL/R, 277
- ABCL/R2, 132
- ABCM, 133
- ABOX, 88
- abstract class, 151, 394, 396–402, 404, 407, 416, 419, 420, 425–427, 432, 437–439, 444
 - relationship, 408, 433
- abstract entity relationship, 398
- ACI-rewriting, 322, 354, 359
- ACORE, 177
- ACT++, 113
- activity tables, 493
- actor
 - behavior startup, 250
 - grammars, 255
 - language, 247
 - specification, 249
 - transitions, 180
- actors, 38, 45, 87, 123, 155, 165, 175–182, 223, 239, 244, 248, 249, 251, 253, 255, 276, 277, 309, 310, 313, 315–317, 352–355, 360, 363, 379, 380, 382, 395, 396, 483, 484
- actorSpace, 7, 8, 15, 19
- ADA, 28, 29, 38, 110
- aggregate, 175–177, 179–182, 184, 186, 187, 189, 190, 192
 - model, 175, 176, 179, 192
 - types, 293
- AKL, 85, 211, 216, 219, 223–229
 - programming system, 224, 229
- ALPS, 110
- Amber, 454
- AND-parallelism, 47, 84, 85, 88
- ANDOR-II, 85
- Andorra Kernel language
 - seeAKL
- Andorra Prolog, 85
- AP1000, 108
- Argus, 206
- Associations, 276
- asynchronous communication, 57, 338, 339, 382
- atomic blocks, 195, 199, 200, 202, 203, 205–207
- atomic function, 291

- atomicity, 197, 200, 207, 291, 306, 482
- autonomous objects, 338
- Avalon, 206

- backtracking, 39, 47
- backtracking process view, 43, 47, 61, 70, 80, 83
- bags, 218, 220
- bandwidth, 13, 14
- barrier, 287, 416, 417
- behavior abstraction, 112
- behavior replacement, 178
- belief, 84
- BiggerTalk, 74
- blackboard system, 258, 276
- bounded buffer, 108, 161, 336, 337, 344
- breadth-first search, 26
- broadcast, 7, 18, 58, 79, 182, 189, 262–264, 266–268, 270–274, 317, 350–352, 383
- Broadway, 16, 17

- C, 5, 285
- C++, 69, 107, 110, 112, 115, 131, 207, 212, 281, 283–290, 292–298, 307–311, 396
- c-future, 465, 466
- C-logic, 77
- CAL, 77
- call/return communication, 4–6
- Camelot, 206
- CANTOR, 177
- capabilities, 60
- CARE, 480, 487, 489–492, 495, 499, 505, 507
- CC++, 112, 281–312
- CCS, 38, 87, 123, 355, 380
- CHAM, 257, 259, 262, 276
 - metaphor, 262
- channels, 82, 217, 218, 262, 266, 306, 367, 488, 489
- CHARM, 278
- Chemical Abstract Machine, 380
 - see CHAM
- Choices, 393–451
 - applications, 446
 - device management, 437
 - frameworks, 401
 - interface, 441
 - message passing system, 418
 - persistent storage, 430
 - process management, 406
 - virtual memory system, 425
- Cholesky Decomposition, 9, 14
- Church-Rosser property, 129, 320, 329
- clauses view, 43, 46, 83
- client-server interaction, 154
- client-server problem, 239
- closed object model, 69
- closed systems, 23
- Clouds, 207
- CLP, 82

- clustering, 502
- CM-5, 108
- commit operator, 50
- committed choice non-determinism, 28–30, 45
- committed choice process view, 43, 47, 60, 69, 78, 83
- Common Lisp, 207, 362
- communication abstraction, 4, 5, 8
- communication media, 213
- component, 22–42, 151, 163, 217, 219, 257, 278, 393, 394, 401, 402, 404, 406, 418, 425, 430, 438, 480, 487–489, 491, 495–497, 499
 - models, 28
- component-based, 22, 23, 26–28, 31, 32, 34, 35, 37–39
 - design space, 33, 38, 39
 - logic programming, 25
 - models, 23, 26, 31–33, 35, 38
 - software technology, 22, 34, 38–40
- composition
 - dependability protocols, 16, 17
 - function, 164
 - interface, 39
 - module, 39
 - object, 151, 157
 - objects, 152
 - patterns, 167
 - program, 39
 - software, 151, 152, 159, 162, 168
- Compositional C++
 - see CC++
- concrete class, 397, 398, 400, 402, 404, 426, 438
- concurrency profile, 184
- Concurrent Aggregates, 175–177, 181
- concurrent constraint LP, 85
- Concurrent Constraint Programming, 276, 381
- Concurrent Logic Programming, 29–31, 33, 52, 211, 216, 281, 309, 310
- concurrent objects, 155
- Concurrent Prolog, 61
- concurrent rewriting, 317
- Concurrent Smalltalk, 177, 283, 395, 453, 454
- Conditional Rewriting Logic, 276
- conflict matrix, 196
- congruence, 328
- constant delay property, 215
- constant object, 288
- constraint method, 125
- constraint store, 213
- constraint-based languages, 85
- constraints, 276
- constructor, 287
- context future, 465
- context management, 465
- context switching, 410
- continuations, 166, 223, 225, 482, 483, 486
- Cosmic Kernel, 454
- COSMOS, 452–476
 - code distribution, 466
 - memory map, 460
 - structure, 458
 - distribution scheme, 469
 - object naming scheme, 461
 - performance, 469
 - task manager, 463
- counting semaphore, 199
- CPU, 63
- criticalities, 458
- CSSA, 112
- cumulative latencies, 494
- cut-through routing, 488

- D-clause, 64
- data abstraction, 151, 157, 175–177, 179, 181–183, 187, 190, 192, 287, 434
- data parallel, 282
- dataflow languages, 281
- deadlock, 162, 164, 167, 199, 202, 203, 216, 242, 245, 491
- debugging, 39, 46, 168, 225, 283, 357, 358, 453, 454, 475, 482, 491
- declarative programming, 286
- declarativeness, 31, 33, 34
- delegation, 46, 47, 71, 157
- Delta Prolog, 42, 63
- denotational semantics, 378
- dependability, 4, 10, 16, 17, 19
- depth-first search, 26
- device management, 437
- dining philosophers, 11
- direct key specification anomaly, 112
- discrete-event simulation, 300
- dispatcher, 16–18
- Distributed Abstract Data Types, 304
- distributed clause, 64
- distributed logic programming, 56
- distributed objects, 468
- DL, 88
- DLP, 56, 64, 80
- don't care non-determinism, 45
- don't know non-determinism, 25, 26, 29, 30, 33, 37, 45
- double pushout categorical definition, 235, 238
- dynamic dining philosophers, 242

- EAM, 85
- Eiffel II, 127
- Eight Queens, 363
- ELINT, 500–504, 507, 508
- embedded systems, 84
- Emerald, 454
- empiricism, 24, 27, 40, 41
- enabled sets, 113, 115, 116, 118, 156
- encapsulation, 22, 23, 39, 44, 55, 153, 154, 157–159, 162, 277, 287, 434
- Eqlog, 316
- equational logic, 328

- equational simplification, 317
- ESP, 45, 77
- Ethernet, 419
- event goals, 63
- exception handling, 283
- Extended Andorra Model, 85, 88

- F-logic, 77
- fairness, 245, 292, 354, 360, 367
- Fast Fourier Transform, 425
- fault-tolerant communication protocol, 365
- FCP, 310
- FFT, 425
- FGDC, 80
- FIFO buffer, 336
- FIFO scheduling, 407
- filofax object model, 69
- fine-grain computing, 452
- first-order Horn logic, 315
- Flavors, 481
- FOOPLog, 379
- FOOPS, 316, 345, 349, 379
- forums, 264
- frame problem, 315
- framework, 394
- FREEDOM, 77, 85
- functional modules, 317

- G++, 309
- Gamma language, 355
- gang scheduling, 406, 407
- garbage collection, 211–213, 215, 217, 221, 222, 224, 229, 434
- garbage collector, 293
 - copying, 222
 - generational, 222
- Gaussian Elimination, 14
- GHC, 310
- Global Object Namespace, 455
- Global Pointers, 298
- GLU, 363
- grain size, 453
- graph grammars, 235
- guard, 236
- guarded method set, 136

- HAL, 283
- hardware description language, 82
- Harmony, 132
- heap manager, 460
- Hermes, 38
- heterogeneous systems, 362
- history-only sensitiveness, 119, 122, 143
- Horn clause, 44, 47–50
- hot spot, 468, 491
- Hybrid, 152, 157, 159–162, 207
- Hydra, 454
- Hypertext, 38

- I-Structure, 310
- IAM, 257, 259, 261–264, 267, 276–278
 - metaphor, 259, 260, 264, 266
 - multi-agent, 260–262
 - single agent, 258, 259, 277
 - single-agent, 257
- Id, 310
- indeterminism, 29
- inheritance, 42–48, 55, 56, 70, 152–154, 157–159, 162, 197, 206, 212, 277, 288, 315–317, 322, 332, 339, 340, 345–347, 354, 382, 383, 393–395, 397, 523
- inheritance anomaly, 107–109, 112, 113, 115, 119–121, 125–127, 129–132, 140, 146, 147, 315, 337, 344
- initial algebra semantics, 377
- Intensional logics, 84
- inter-object concurrency, 335
- interaction abstract machines
 - see IAM
- interface complexity, 38
- internal concurrency, 153
- interoperability, 167, 168, 195–197
- intra-object concurrency, 335
- intuitionistic logic, 84

- J-Machine, 176, 309, 452, 454, 455, 457–460, 464, 469, 470, 472–475
- Janus, 85, 87, 177, 218
- join continuation, 6
- JOSS, 454

- Kernel Andorra Prolog, 85
- KL-ONE, 88
- knowledge, 84
- KRYPTON, 88

- label/argument translation, 49
- label/predicate translation, 49
- LAMINA, 480, 482, 484, 486, 488, 489, 491, 493, 495–497, 501, 507
 - class, 484, 486
 - computations, 481
 - object model, 481, 483, 487
- latency, 453
- LIFE, 82, 85
- Linda, 8, 38, 86, 164, 276, 363
- Linear Logic, 80, 84, 266, 380
- linear logic programming, 381
- Linear Objects, 80, 89, 257, 266
- linearizability, 197
- lock, 154, 181, 196, 200, 205, 207, 264, 416
- logic programming, 22, 26, 28, 31, 42, 47, 177, 211, 212, 215, 219, 226, 277, 299, 310, 317, 368–370, 379, 381
- logical processor objects, 295
- LOGIN, 42, 77, 82, 88
- LP-completeness, 22, 35
- Lucid, 84

- Lucy, 85, 87, 218
- Mach, 206, 394
- MacObject, 50
- mailbox, 5, 63, 156
- Maude, 87, 128, 129, 140, 313
- MaudeLog, 316, 368
- MDP, 452
- Mediators, 110
- Medusa, 454
- megaprogramming, 38
- MELD, 195, 197, 198, 200, 202, 203, 205–208
- MELDC, 208
- memory management, 428, 452, 453, 457–459, 474
- memory map, 459
- message passing system, 418
- message-driven processor, 452
- meta-actor, 16, 17
- meta-interpretation view, 43, 48, 56, 70, 81–83
- metalogical predicates, 44
- metaobject, 131, 350–352
 - protocol, 352
- method guards, 118
- method sets, 134
- MIMD, 361
- modal logic, 84
- Model/View/Controller, 397
- modeling complexity, 38
- modularity, 3, 4, 13, 15, 19, 20, 44, 151, 176, 178, 182, 184, 190, 192, 316, 379, 382
- module inheritance, 345
- module inheritance, 344
- monitoring, 39, 168, 479–481, 486, 489, 499, 507
- monitors, 306
- monotonicity, 211, 214, 221
- Mosaic C, 176
- multi-actor constraints, 10, 11
- multi-agent systems, 85
- multi-object constraints, 10
- multi-sets, 218, 220, 257, 260, 276
- multilingual programming, 285
- Multilog, 64
- multiple inheritance, 55, 343, 344
- multiple writers, 217
- multiway merger, 215, 216, 218
 - constant delay, 216
- Muse, 396
- mutable data, 292
- Mutual References, 217
- N-body simulation, 484
- name manager, 461
- natural language processing, 87
- nested transactions, 206
- network load, 474
- neural nets, 28
- NLC grammars, 238
- NP-completeness, 22, 35, 37
- O-logic, 77
- OBJ, 316, 362
- OBJ3, 82, 317, 346
- object
 - autonomy, 153
 - calculus, 151, 152, 162, 163, 167, 168
 - clause, 59
 - creation, 349
 - deletion, 349
- object flow diagram, 409
- object-oriented databases, 88
- object-oriented modules, 330, 334, 359
- ObjVProlog, 56, 70
- observability, 22, 27, 35
- OC, 163
 - see object calculus
- open object model, 69
- Open Path Expressions, 88
- open systems, 23, 257, 262, 264, 277, 365
- operational semantics, 378
- Operations Research algorithm, 267
- optimistic concurrency control, 203, 206
- OR-parallel Prolog, 82
- OR-parallelism, 47, 85, 88
- order-sorted
 - algebra, 318, 323, 381
 - equational logic, 318, 369, 379
 - Horn logic, 369
 - rewriting logic, 369
 - semantics, 346
- OTM, 112
- overloading, 288
- P-Prolog, 85
- Pandora, 85
- panels, 497
- Papyrus architecture, 196
- paradigmatic incompatibility, 22, 26–28
- parafunctional programming, 303
- parallel blocks, 289, 290, 295, 302, 310
- Parallel NU-Prolog, 85
- parameterized functional modules, 320
- Parlog, 50, 61, 78, 81, 219, 310
- Parlog Programming System, 81
- Parlog++, 55
- partitioned name space, 44
- partitioned predicate space, 44
- partitioned state, 182
- Pascal, 5
- Path Expressions, 112
- pattern language, 162
- pattern-directed communication, 5, 7
- PCE, 42
- PCN, 281, 283, 309–311, 363
- persistent storage, 395, 426, 430, 432–437, 441, 447
- Petri nets, 327, 333, 355, 375, 378, 383

- Phocus, 73
- placement strategy, 10, 14, 19
- platform, 243
- plug-compatibility, 167
- Polka, 61, 79, 211, 219
- Polka-B, 80, 86
- polymorphism, 42, 44, 158, 246, 288, 394
 - inclusion, 394
- POOL, 85, 147
- POOL-I, 159
- POOL-T, 177
- POOL/I, 110, 147
- POOL/S, 147
- POOL/T, 107, 110
- ports, 211
- Poset, 376
- Post systems, 355
- PPS, 81
- PRAM, 211, 225, 227
 - EREW, 212
- primary method, 125
- probes, 495
- process management, 406
- process semantics, 87
- process view, 47, 82, 83
- Procol. 107. 112
- producer-consumer, 200
- Program Composition Notation
 - see PCN
- program transformation, 119
- program verification, 22, 36, 286
- Prolog, 42, 50
- prolog, 24–26, 29, 30, 38
- Prolog++, 50, 74
- Prolog-D-Linda, 86
- Prolog/V, 42
- property-driven communication, 262
- ProWindows, 42
- PSI Machine, 45
- public key encryption, 82

- quasi-concurrent objects, 155
- Quintec-Prolog, 50

- R ground Church-Rosser*, 330
- R-homomorphism*, 374
- R-normal form*, 329
- rationalism, 23, 24, 27, 32, 40, 41
- reactive systems, 281–284
- reactiveness, 22
- read-only variable, 61
- real-time systems, 84, 196, 500
- reflection, 16, 17, 56, 317, 351, 352, 377, 395, 396
 - functor, 376
- reflexivity, 328
- remote procedure call, 6, 157, 311, 312, 395, 396, 420, 447
- remote procedure calls, 309

- replacement, 328
- replication, 8, 17, 501
 - consistent, 181
 - loosely consistent, 181
- reply scheduling transparency, 154
- request scheduling transparency, 153
- retract operators, 351
- retraction, 236–238, 242
- Rewrite Rule Machine, 354, 358, 362
- rewriting logic, 315, 326
- Rosette, 116
- round robin scheduling, 407
- rule inheritance, 345
- rules of deduction, 328

- S-expressions, 115
- Sandra, 69
- scheduling, 153–157, 159, 160, 224, 352, 396, 406, 410, 411, 413, 414, 442, 452, 455, 457, 465, 479, 482–484, 487, 490, 494, 508
- self, 71, 74, 80, 139
- semantic complexity, 38
- semaphore, 416
- sentences, 327
- sequential block, 200, 290
- serializability, 196
- serializable transactions, 196, 202
- serialization, 175–180, 182, 183, 185, 187, 188, 190, 192
- services
 - global, 458
 - local, 457
 - relocatable, 458
- shared logical variables, 47, 62
- Shared Prolog, 86, 276
- SIMD, 361
- SIMPLE, 480, 489, 490, 497, 499
- Simple Maude, 356
- Simplex, 425
- SIMPOS, 45
- Simula, 69, 151
- Simula67, 55, 313
- SINA, 112
- single-assignment variables, 286, 309
- SISCTus Prolog, 74
- Smalltalk, 42, 55, 56, 69, 107, 151, 212
- Smalltalk-80, 110, 139, 454
- Smalltalk/V, 56
- SML, 146
- software complexity, 22, 34–37
- sort constraints, 321
- SP-Object, 56
- SP-Prolog, 56
- spawn, 289, 290
- SPOOL, 68
- SR, 110
- state modification anomaly, 143
- state partitioning anomaly, 142

- store-and-forward routing, 488
- Strand, 281, 283, 310, 311, 363
- stream
 - hiding, 60
- streams, 211, 215, 216, 218–224, 226, 227, 481
 - distributor, 226, 227
 - end of, 215–217, 222
 - merging, 215, 216
 - shared, 217
- strip chart, 492
- strong typing, 157, 283, 287
- structured cooperation, 182
- subforums, 264
- subframework, 397
- subsort inheritance, 322
- subtransaction, 206
- super, 71, 73, 139
- suspendable message, 61
- Swarm, 276
- symmetry, 328
- sync
 - data, 292
 - objects, 294
 - types, 294
- synchronisation idioms, 223
- synchronization constraints, 6, 8–10, 16, 39, 108, 110, 112, 120–123, 127, 128, 130, 133, 134
- synchronizers, 10–13, 137, 141, 142, 144
- synchronizing actions, 124, 126
- synchronous communication, 338
- system transitions, 180

- T-Prolog, 42, 61
- tail forwarding, 472
- task management, 455
- TBOX, 88
- templates, 283
- temporal logic, 84
- Tera, 309
- term classes, 57
- THLP, 84
- thread package, 308
- timestamps, 196
- trace-only sensitiveness, 119
- transaction, 69, 132
 - blocks, 202
 - distributed, 132
 - group, 207
- transduction, 39
- transition
 - method, 125
- transition specifications, 137, 141–144
- transitivity, 328
- transparent message-distribution, 217
- triggers, 243
- tuple space, 164
- Tuple Space Smalltalk, 112
- Turing machines, 355

- two-phase commit, 203
- two-phase locking, 196
- type casts, 288
- types, 158, 283

- Unidraw, 397
- unification, 25, 33
- Unity model, 276, 281, 355
- UNIX, 260, 394, 453
- utilization histograms, 491

- V, 394
- virtual function table, 134
- virtual memory system, 425
- volatile object, 288, 298
- Vulcan, 79, 82, 211, 219

- Warren Abstract Machine, 76
- weakly open systems, 23

- X-grammar, 235, 239
- XO/R, 131

- Y-grammar, 235, 239