
Index

- O , **16**
- Ω , **16**
- $\cap$, **16**
- $\cup$, **16**
- $\oplus$, **188**
- $\#$, **16**
- $+$, **23, 55**
- $-$, **23, 55**
- $*$, **23, 55**
- $/$, **23**
- $^$, **23**
- $\&$, **55**
- $\dots$, **55**
- $\{ \}$, **55**
- $[]$, **56**
- $M\{x\}$, **56**
- $:=$, **24, 86, 155**
- $\rightarrow$, **120**
- $\neg$, **14**
- $||$, **14**
- 0-1 knapsack, **99**

- abstract data type, **151, 184**
- Ackermann function, 103, **105, 115**
- Ada, 32, 51
- add, **56**
- adjacency-list representation, **69**
- adjacency-matrix representation, **69**
- aggregate expression, 50, **57**
- Aho, 21
- ALGOL, 51
- alias, **153, 160, 165, 204**
- alias analysis, **160, 204, 211**
- all, **57**
- Allen, 20, 51
- Alloy, 21
- Analytical Engine, 32, 51

- anchor of auxiliary map, **126**
- Andersen, 149
- APL, 51
- APTS, 81, 211
- argument, **13, 119**
- array, **15, 23**
- array, **86**
- array construction, **49, 86**
- array element access, **86**
- aspect-oriented programming (AOP), 186
- assignment command, **24, 155**
- associative access, **67, 68, 130, 132–134, 139**
- asymptotic complexity, **16**
- asymptotic improvement, **12**
- asymptotic lower bound, **16**
- asymptotic upper bound, **16**
- attribute closure, 81, 149
- attribute-based access control, 156
- auxiliary map, **73, 78, 125**
- auxiliary relation, 121, 122, 140, 141
- auxiliary space, **139**
- auxiliary specialization, **190, 198**
- auxiliary value, **29, 39, 95, 189, 194, 209, 211**

- B, 21
- Babbage, 20, 32, 51
- Backus, 114
- base, **68, 131, 133, 203**
- base-case condition, **90, 91, 97**
- based representation, **68, 81, 130, 132, 134, 149, 203**
- BASIC, 51, 113
- basic data, **201**
- Bauer, 20, 51

- Beeri, 149
 Bernstein, 21, 149
 binary decision diagram (BDD), 148
 binding, **14**
 binding expression, **86**
 binomial coefficient, **88**
 Bird, 20, 113
 Blaine, 20
 Boolean value, **23**
 bottom-up, 148, 149
 bound, **14**
 Boyle, 20
 Brayman, 211
 breadth-first search (BFS), **70**
 Briggs, 32
 Broy, 20
 Burstall, 113, 114

 C, 51, 81
 C++, 51, 184, 185
 cache-and-prune, **192**, 193, 199, 211
 Cachet, 211
 Cai, 20, 81
 CAML, 113
 chain rule, 62, 93, 171, 212
 change propagation, 20, 62, 208
 CheckAccess query, **72**
 Chin, 20, 113
 CIP, 20, 114, 211
 class definition, **153**
 Cocke, 20, 51
 code, **13**
 Cohen, 113
 command, **24**
 comment, 14
 Common Lisp, 113
 compile time, **14**
 compiler, **14**
 complexity, **16**
 complexity class, 16
 component of tuple, **56**
 composition, 62, 64, 186
 compound operation, **14**
 compound type, **14**
 computer program, **13**
 conclusion of rule, **120**
 conditional command, **24**
 conditional expression, **86**
 Constable, 51
 constant complexity, **16**
 constraint programming, 208
 constructor application, **85**
 continuation, 113
 continuity, 9, 31, 93

 control abstraction, **6**
 copy elimination, 81
 Cormen, 21, 115

 Danvy, 116
 Darlington, 113, 114
 data abstraction, **6**
 data locality, 206
 data structure, **15**
 data type, **14**
 database programming, **6**
 Datalog, 81, 117–119
 dead, **65**, 66, 127, 191, 197, 198
 dead code, 78, 127, 190, 198, 203
 debugging, 182, 186
 decidable, **17**, 204, 205
 decision problem, **16**
 declarative language, **6**
 decrement operation, **89**, 91
 deductive synthesis, 20
 def, **85**, **154**
 deforestation, 20, 113
 del, **56**
 depth-first search (DFS), **70**
 dequeue, **15**
 derivative, **30**, 62, 212
 derived data, **201**
 Dershowitz, 20
 design pattern, 168, 185
 DFA minimization, 81
 Difference Engine, 20, 32, 51
 differentiation, 9, 30, 31, 93, 115, 210
 Dijkstra, 20
 divide and conquer, 207
 dom, **56**
 domain, **132**
 domain of map, **56**
 dominated convergence, 20, 80
 doubly linked list, **15**
 dynamic, **15**
 dynamic programming, 51, **83**, 93, 114, 115

 Earley, 20
 edit distance, **87**
 EHR, 172
 else, **24**, **86**
 ENIAC, 31, 41, 51
 enqueue, **15**
 equal card, **121**
 equality reasoning, 190, 198, 204
 error, 11
 existentially quantified expression, **57**
 exists, **56**, **57**

- expand, 96, 97, 190
 exponential complexity, **16**
 expression, **23**
 extended regular-expression pattern, **183**
- fact, **120**
 factorial, **104**, 115
 false, 23
 Fibonacci function, 80, 88, 108, 116
 Fibonacci number, 104, **108**, 115
 Field, 18
 field, **15**, **153**
 field access expression, **154**
 field assignment command, **155**
 field definition, **153**
 filtering, 148
 finite differencing, 20, 80
 firing of rule, **122**
 fixed point of function, **59**
 floating-point number, **23**
 Flora, 185
 Fong, 20
 for loop over integers, **24**
 for loop over sets, **57**
 FORTRAN, 51, 113
 FP, 114
 free variable, **14**
 function, **13**
 function application, **86**
 function call, **86**
 function definition, **85**
 function symbol, 208
 functional programming, **6**
 functor, 208
 fusion, 20, 113
- Ganzinger, 148
 global variable, **14**
 Goldberg, 20, 116
 Goldstine, 51
 Gorbovitski, 81, 186
 Goyal, 20, 81, 82
 graph query, 3, 184
 graph reachability, 3, **54**, 59, 69, 113
 greedy algorithm, 115, 208
 Gries, 20, 41, 51
- hash table, **15**
 hashing, **16**
 Haskell, 80, 113
 heap, 172, 205, 208
 Heintze, 149
 higher-order function, 208
 Hoover, 18
- Hopcroft, 21, 82
 Horwitz, 18
 Hu, 20, 113
 hypothesis of rule, **120**
- identifier, **14**
 if, **24**, **86**, **127**
 III, **5**
 image blurring, 2, 25, 41
 image under map, **56**, 127
 image union under map, **59**
 imperative programming, **6**
 in, **55**, **86**
 in-degree, **136**
 increment operation, 91
 incremental version, **18**, **188**
 incrementalization, **7**, **22**
 incrementalization rule, 63, **72**, 74, 162, **163**
 core form, **163**
 general form, **169**
 induction, 20, 90
 inferred, **120**
 initial argument, **90**, 97
 instrumentation, 182, 186
 integer, **23**
 integer division, 26
 integer square root, 26, 36
 integration, 9, 93, 115, 210
 intermediate result, **29**, 65, 94, 95, 189, 192, 209, 211
 invariant, **27**
 checking, 181
 discovering, 180
 maintaining, **27**
 invariant rule, **178**
 inverse map, **59**, 73, 77
 InvTS, 211
- Java, 51, 184, 185
 Jha, 149
 Johnson, 51
 join order, 75, 82
 join query, 76
 Jones, 113
 Jung, 21
- Katz, 51
 Kennedy, 20, 51
 key of hash table, **15**
 KIDS, 20, 114, 211
 Kifer, 21
 Kleinberg, 21
 Koenig, 20
 Kohn, 210, 211

-
- language construct, 11
 - LCS, **84**
 - Leiserson, 21, 115
 - let, **86**
 - Lewis, 21
 - Li, 52
 - linked list, **15**
 - Lisp, 113
 - literal, **23**
 - Liu, 20, 21, 51, 81, 82, 114, 148, 150, 185, 211
 - local variable, **14**
 - logic programming, **6**
 - logic rule, **6**, 119
 - longest common subsequence, 3, **84**
 - loop command, **24**
 - loop variable, **24**
 - loops and arrays, **6**
 - Lovelace, 32

 - magic-sets, 117, 148
 - Manna, 20
 - map, **56**
 - match a tuple pattern, **57**, 127
 - materialized view, 20
 - McAllester, 148
 - memoization, 20, 114
 - meta-variable, **164**
 - method, **153**
 - method definition, **153**
 - method invocation command, **155**
 - method invocation expression, **154**
 - Michie, 20
 - minimum increment, **89**
 - Miranda, 80, 113
 - ML, 113
 - model checking, 81
 - module abstraction, 7, **151**
 - Moore, 20
 - multi-map, **56**
 - Mycroft, 115

 - name, **14**
 - name-reduction closure, 145, 149
 - Nerode, 115, 210, 211
 - new, **154**
 - next*, **89**
 - non-anchor of auxiliary map, **126**
 - nondeterministic polynomial time, **17**
 - not in, **55**
 - NP, **17**
 - NP-complete, **17**
 - null, **14**
 - Nuprl, 51

 - O’Leary, 51
 - object, **153**
 - object class test, **154**
 - object creation command, **154**
 - object graph, **183**
 - object-oriented programming, **6**
 - objects and classes, **6**
 - observer pattern, 168, 185
 - OCaml, 113, 185
 - OCL, 80
 - of, **127**
 - on demand, **33**, 150
 - optimization, 11
 - OQL, 185
 - order-of-magnitude improvement, **12**
 - output space, **139**

 - P, 16**
 - Paige, 20, 80–82
 - parameter, 7, 72, **85**, **153**, **159**
 - partial evaluation, 113, 148
 - partition refinement, 81
 - Partsch, 20
 - Pascal, 51
 - pattern variable, **57**
 - Pettorossi, 113, 115
 - pointer, **14**
 - pointer analysis, 81, **142**, 149
 - polynomial complexity, **16**
 - polynomial time, **17**
 - pop, **15**
 - power set, **61**
 - predicate, **13**, 119
 - prefix firing, 148
 - primitive function, **85**
 - primitive operation, **14**
 - primitive recursive, 105
 - primitive type, **14**, 85
 - Priyalakshmi, 21, 81
 - program construct, **13**
 - program schematology, 114
 - program segment, **13**
 - Proietti, 115
 - Prolog, 80, 117, 147
 - promotion and accumulation, 20, 113
 - pseudo-polynomial time, **100**
 - Pugh, 18
 - push, **15**
 - Python, 51, 80, 81, 185, 211

 - quantified expression, **57**
 - query, 1, 72, **157**, 163
 - query optimization, 76
 - queue, **15**

- ran, **56**
- range of map, **56**
- RBAC, **71**, **75**, **76**, **81**
- real-time simulation, **80**
- record, **15**
- recursion removal, **114**
- recursive call, **83**
- recursive data structure, **85**
- recursive function, **6**, **83**, **86**
- red-black tree, **172**
- refactoring, **182**
- reference, **14**
- regular expression, **81**, **183**
- regular path expression, **183**
- regular path query, **81**, **144**
- relation, **13**
- relational calculus, **76**, **81**
- Reps, **17**, **149**
- result, **13**
- result map, **72**
- Rete algorithm, **148**
- return command, **87**, **154**
- return value, **13**
- reverse engineering, **180**, **181**
- Reynolds, **20**
- Rivest, **21**, **115**
- Robocode, **176**, **177**, **185**
- role-based access control (RBAC), **71**
- Rothamel, **20**, **21**, **81**, **185**
- rule, **6**, **119**
 - Datalog rule, **119**
 - simplified form, **121**
- runtime, **15**
- runtime check, **160**
- runtime invariant checking, **182**, **186**
- runtime verification, **181**

- Scheme, **113**
- Scherlis, **113**
- Schonberg, **20**
- Schwartz, **20**
- scope, **14**
- Scott, **21**
- selective caching, **211**
- selector application, **85**
- self, **154**
- self object reference expression, **154**
- semantics, **13**
- semi-naive evaluation, **148**
- sequencing command, **24**
- set, **13**, **55**
- set comprehension, **56**
- set expression, **6**
- SETL, **80**

- Sharir, **20**
- shift, **2**, **28**, **30**, **35**, **36**
- short-circuit semantics, **128**
- single-source shortest path, **102**
 - Bellman-Ford algorithm, **103**
- singly linked list, **15**
- size, **16**
 - argument size, **136**
 - domain size, **136**
 - relation size, **136**
 - relative argument size, **136**
 - size of set, **55**, **155**, **164**
- Smalltalk, **80**
- Smith, **20**, **186**
- SML, **113**
- sorting, **196**
 - insertion sort, **196**
 - merge sort, **198**
 - selection sort, **197**
- sparse, **69**, **82**, **135**
- SPKI/SDSI, **145**, **149**
- SQL, **80**, **113**
- stack, **15**
- state, **14**, **24**, **154**, **168**, **175**, **177**
- statement, **24**
- static, **14**
- Step Implement, **5**
- Step Incrementalize, **5**
- Step Iterate, **5**
- stock averages, **26**
- Stoller, **20**, **114**, **148**
- stream processing, **209**
- strength reduction, **20**, **28**, **51**, **189**
- strengthening loop invariants, **20**, **51**
- strict operation, **62**
- strongly based, **68**, **131**
- strongly connected components, **208**
- subset, **55**
- syntax, **13**
- Szeliski, **51**

- tabling, **117**, **148**
- tabulation, **20**, **114**
- tail recursion, **87**
- Tardieu, **149**
- Tardos, **21**
- Tarjan, **82**
- Teitelbaum, **17**
- Tekle, **150**
- tester application, **85**
- then, **86**
- top-down, **148**
- Tower of Hanoi, **104**, **110**, **115**
- transitive closure, **118**, **119–121**, **124**,

240	<i>Index</i>
126, 128, 129, 134, 136–138, 140	Waldinger, 20
tree pattern matching, 81	Wand, 113
true, 23	Waters, 20, 51
trust management, 144, 149	weakly based, 68, 131
tuple, 13, 56	Webber, 113
tuple pattern, 57 , 127	Wegbreit, 113
tupling, 20, 113, 115	where, 85
Ullman, 20, 21	while loop, 24
undecidable, 17 , 204, 205	wild card, 121
unfold, 190	Willard, 81, 149
universally quantified expression, 57	work-set, 61, 127
update, 7 , 72, 157 , 163	XML, 80, 186
variable, 14	Xquery, 80
variable assignment command, 24	XSB, 148
verification, 181	Yang, 115
view maintenance, 20, 80	Yap, 148
VLSI, 36	Z, 21, 80, 81
Wadler, 20, 113	Zabih, 51
	Zhang, 52