# Index


**A**
A-list, 113
Abstract machine, 1
Ackermann
  function, 148
Activation record, 94
  dynamic chain pointer, 105
  for function, 97
  for in-line blocks, 96
  for procedure, 97
  static chain pointer, 97
Ada, 427
Ada Byron Lovelace, 427
Aiken, 414
Algol, 215, 418
Alias, 71
Aliasing, 68, 71
Analysis
  lexical, 41
  semantic, 44
  syntactic, 43
APL, 124
Applet, 429
Application, 335, 338, 339
Arithmetic
  pointer, 224
Array, 216
  address calculation, 218
  allocation, 218, 219
    dope vector, 220
  bounds checking, 217
  shape, 219
  in C, 227
  Java, 221
  multidimensional, 216
  slice, 217
ASCC/MARK I, 414
Assembly language, 415
Assignment, *see* Command
Association
  creation, 75
  deactivation, 75
  destruction, 75
  environment, 69
  lifetime, 76
  reactivation, 75

**B**
Babbage, C., 427
Backtracking, 393
Backus, J., 417
Base class
  fragile, 305
Beta, 339
Binding
  deep, 179
  environment, 69
  shallow, 179
Block, 70, 137
  activation record, 96
  and procedure, 72
  in-line, 72
  protected, 187
BNF, 419
Böhm, 139
Boolean, 204
Bytecode, *see* Java

**C**
C, 421
  array, 216
  call by reference, 172
  enumeration, 208



M. Gabbrielli, S. Martini, *Programming Languages: Principles and Paradigms,* Undergraduate Topics in Computer Science,
DOI 10.1007/978-1-84882-914-5, © Springer-Verlag London Limited 2010

environment, 183
union, 212
C++, 426
call by reference, 242
class, 286
dynamic method dispatch, 300
inheritance, 292
subtype, 288
template, 240
virtual member function, 290
Call
of procedure, 100
Case, *see* Command
Cast, 235, 236
non-converting, 236
Central Reference Table, 111
Character, 204
CHIP, 428
Chomsky, N., 29
Church, A., 63
Church's Thesis, 63
Class, 283
abstract, 289
derived, 288
fragile base, 305
implementation, 301
Clause, 376
as procedure, 388
body, 377
definite, 376
head, 377
unit, 377
Closure, 176, 181, 183, 184
CLP, 428
COBOL, 420
Coercion, 235
Colmerauer, 428
Command, 129
`;`, 138
assignment, 131
assignment in C, 132
block, 137
`case`, 141
composite, 136
conditional, 140
syntactic ambiguity, 140
`do`, 145
for, 145
`goto`, 137
`if`, 140
iteration
bounded, 144, 145
unbounded, 144
iterative, 144
`repeat`, 144
sequence, 136
structured, 151
while, 144
Compatibility of types, 234
Compile time, 69
Compiler, 1, 41
Computability, 60
Computation, 45
Computed answer, 392–394
Constraint, 427
Constructor, 209, 229
type, 230
Continuation, 159
Contravariant, 324
Copy rule, 175, 337
Covariant, 324
CRT, *see* Central Reference Table
CSP, 405
Curry, Haskell, 346


**D**


Dahl, 420
Dangling pointer, 247, 248
Datalog, 403
Deallocation, 225, 247
Declaration, 70
incomplete, 85
scope, 85
Deep binding, 179
Delegation, 286
Denotable, 201
Denotable object
access, 76
creation, 76
destruction, 76
lifetime, 76
modification, 76
Dereference, 223
l-value, 222
Derivation, 32
Dijkstra, E., 161
Discriminant, 212
Dispatch, 297
multiple, 301
Dope vector, 220
Dotted pair, 121
Downcast, 304
Dylan, 286
Dynamic chain, 97
pointer, 97
Dynamic selection, 283

**E**
Eckert, 413
EDSAC, 413
EDVAC, 414
Emulation, 9, 10
Engine
  analytic, 427
  difference, 427
ENIAC, 414
Enumeration, 207
Environment, 68
  and block, 70
  global, 73
  local, 73
  non-local, 73
  referencing, 70
Equality
  of terms, 382
Equation
  between terms, 377
Equivalence
  by name, 231
  of types, 201, 230
  structural, 232
Evaluation
  applicative-order, 342
  by name, 342
  by need, 343
  by value, 342
  eager, 342
  innermost, 342
  lazy, 343
  leftmost, 342
  normal order, 342
  outermost, 342
  strategy, 341
Exception, 186
  implementation, 190
  resumption after, 188
  rethrowing, 191
Expressible, 201
Expression, 119
  associativity, 123
  evaluation, 123, 125
    APL, 124
    eager, 127
    lazy, 128
    short circuit, 128
    tree, 126
  infix, 126
  notation
    Cambridge Polish, 122
    infix, 120
    Polish, 121
    postfix, 122
    prefix, 121
  precedence, 123
  prefix, 124
  representation as tree, 123
  semantics, 49
  syntax, 120
Expressiveness
  of bounded iteration, 147

**F**
Fibonacci
  function, 153
  sequence, 92
Field
  of record, 209
  shadowing, 289
Firmware, 10
First-order Logic, 374
Fix, 205
Fixpoint, 361, 362
Fixed point, 205
Float, 205
Floating point, 205
For, *see* Command
FORTRAN, 416, 417
Fragile
  base class, 305
Fragmentation, 103
  external, 103
  internal, 103
Funarg problem, 193
Function, 166
  activation record, 97
  as parameter, 179
  as result, 184
  computable, 60
  covariant, 324
  higher-order, 178
  partial, 12
  virtual in C++, 290
Futamura, projection, 25

**G**
Garbage collection, 250
  copying, 255
  mark and compact, 255
  mark and sweep, 253
  pointer reversal, 254
  reference counting, 251
  stop and copy, 255
Gauss-Jordan, 404

Generic, 314, 317
implementation, 321
Goal, 377
evaluation, 389
Gödel, K., 61
Gosling, 428
Goto, *see* Command
Grammar
ambiguous, 37
context-free, 30
contextual, 39
regular, 42

**H**
Halting Problem, 57
Handler, 187
Hanoi
towers of, 397
Hardware
implementation, 9
Haskell, 202, 334
Heap, 101
block
fixed length, 101
variable length, 103
buddy system, 105
compaction, 104
Fibonacci, 105
fragmentation, 103
external, 103
internal, 103
management, 101
multiple free list, 105
single free list, 104
Herbrand, J., 370
Herbrand Universe, 382
Hiding
of information, 265, 268
Hierarchies of abstract machines, 21
Higher order, 178, 364
Hoare, C.A.R., 418
Hopper, 420
HTML, 428

**I**
Ichbiah, 427
If, *see* Command
Implementation, 9, 28, 265
compiled, 13
of Pascal, 20
purely compiled, 14
purely interpreted, 13, 14
real case, 17
Index
type of, 216
Inductive definitions, 153
Inference
type, 201
Inheritance, 292
implementation, 292
interface, 292
multiple, 309
with replication, 311
with sharing, 312
Integer, 205
Interface, 268
Interpreter, 1, 2, 7, 14
Interval, 203, 208

**J**
Jacopini, 139
Jaffar, J., 428
Java
array, 323
bytecode, 429
generic, 317
inheritance, 294
interface, 288
subtype, 287
wildcard, 322, 323
Java Virtual Machine, 306, 307, 429
Javascript, 286
Jensen
device, 178

**K**
Kay, A., 420, 423
Kleene, S., 61
Kowalski, 425

**L**
L-value, 132, 209
Lambda calculus, 358
Landin, Peter, 161, 353
Language
declarative, 137
assembly, 5
formal, 30
functional, 345
high-level, 5
higher order, 178
imperative, 137
logic, 376
low-level, 5
Lifetime, 76, 168, 220

LISP, 112, 121, 419
garbage collection, 250
side effects, 351
List
Prolog, 371
Locks and keys, 249
Logic program, 376
computational model, 387
declarative interpretation, 388
failure, 390
procedural interpretation, 388
success, 390
Logic programming, 369
constraint, 404
declarative interpretation, 388
procedural interpretation, 388
Lukasiewicz, L., 121

**M**
Machine
abstract, 1
hardware, 3
implementation, 9, 18
intermediate, 17, 20
Java, 20
Pascal, 20
interpreter, 1, 2
memory, 6, 8
SECD, 353
von Neumann, 413
MARK I, 414
Mauchly, 413
McCarthy, J., 419
Memory, 91
heap, 101
semantic domain, 135
stack, 93
static, 93
Menabrea, 427
Method
binary, 327
dynamic lookup, 297
redefinition, 288
static, 285
Method overriding, 288
Mix-in, 294
ML, 334, 335, 346, 347, 350, 424
recursive type, 227
reference cell, 350
Module, 271
Monomorphic, 237
Multimethod, 301

**N**
Name, 67
Name clash, 293
Name conflict, 295
Naur, 418
Negation, 401
as failure, 402
Non-determinism, 392
Nygaard, 420

**O**
Object
allocation, 286
creation, 76
deallocation, 286
denotable, 67, 69
access, 76
destruction, 76
lifetime, 76
modification, 76
implementation, 303
Operator
associativity, 123
precedence, 120
Order
column, 218
row, 218
Ordinal, 212
Overloading, 197, 238
Overriding, 288

**P**
P-code, 422
Package, 271
Paradigm
functional, 333
logic, 369
object-oriented, 277
Parameter, 166, 169
actual, 166
call by constant, 171
call by name, 174
call by reference, 170
in C, 172
call by result, 172
call by value, 169
call by value-result, 173
formal, 166
passing, 169
Parser, 43
Pascal, 199, 422
equivalence by name, 231
variant record, 211
Pattern matching, 347

Peano, G., 153
Pointer, 222
Pointer dereference, 248
Pointer reversal, 254
Polymorphism, 314
  explicit, 240
  implicit, 240
  inclusion, 238
  parametric, 238
  subtype, 238
  universal, 238
  universal parametric, 239
  universal subtype, 241
Predicate, 371, 375
Predicate calculus, 374
Procedure call
  calling sequence, 100
  epilogue, 100
Production, 30
Program transformation, 23
Prolog, 369, 398, 424
  arithmetic, 399
  backtracking, 393
  clause selection, 392
  computational model, 387
  control, 392
  cut, 400
  disjunction, 401
  failure, 395
  history, 424
  negation, 401
  non-determinism, 393
  occurs check, 385
  selection rule, 390
Prolog II, 428
Prolog III, 428

**Q**
Query, 377

**R**
R-value, 132
Real, 205
Reals, 205
Record, 209
  variant, 211, 212
Recursion, 91, 93
  tail, 91, 155
Redex, 339
Reduction, 337
Reference counter, 251
Refutation
  SLD, 392
Representation independence, 270
Resolution, 369, 376
  SLD, 376, 392
Rewriting, 334, 337
Ritchie, 421
Robinson, 425
Robinson, A., 370
Root set, 253
Roussel, 425
Rule
  beta, 339
  copy, 175, 337
  visibility, 67
Runtime, 69

**S**
S-expressions, 121
Scheme, 202, 229, 247
Scope, 77
  A-list, 113
  Ada, 83
  C, 83
  CRT, 111
  declaration, 78
  display, 109
  dynamic, 80
  dynamic chain, 97
  holes, 82
  implementation, 105
  Java, 78
  Pascal, 82
  problems, 82
  rules, 77
  static, 78
  static chain, 97
SECD, 353
Selection rule, 390
Self, 286
Semantics, 46, 48
  denotational, 46
  operational, 46
  static, 40
Shadowing, 289
Shallow binding, 179
Side effect, 127
Simula, 420
  class, 284
Slice, 217
Smalltalk, 423
Solution
  of constraints, 404
Spaghetti code, 151
Specification
  of ADT, 270

Static
in C, 168
Strategy
evaluation, 341
Stream, 349
Strongly typed, 201
Stroustrup, 426
Subclass, 288
Substitution, 379
application, 380
composition, 380
ground, 379
renaming, 381
restriction, 381
most general, 381
Subtype, 235, 287
Syntactic sugar, 121, 161
Syntax, 27, 28
abstract, 43
System stack, 91

**T**
Tag, 212
Tail recursion, 91, 155
Template, 242
Term, 375
ground, 375
Theorem
of Böhm and Jacopini, 138
Thompson, 421
Thunk, 182
Tombstone, 248
Towers of Hanoi, 397
Transition, 47, 48
Tree, 33
derivation, 33
syntax, 125
Turing complete, 61
Turing equivalent, 61
Turing machine, 18, 60
Type, 197
base of interval, 208
checker, 199
checking, 202, 244
compatibility, 234
composite, 209
conversion, 234
discrete, 209
equivalence, 230
explicit conversion, 237
index, 216
inference, 244
monomorphic, 237
of functions, 229
recursive, 227
recursive in ML, 230
scalar, 203
simple, 203
void, 206
Type checker, 233
Type safety, 197, 201, 202, 213, 218, 226
Type system, 197
Typing
dynamic, 202
static, 202

**U**
Undecidability, 59
of halting problem, 59
of static type checking, 203
Unification, 377
algorithm, 383
Martelli and Montanari algorithm, 384
theory, 377
Unifier
most general, 381

**V**
Value
denotable, 137, 201
expressible, 137, 201
functional, 340
storable, 137, 201
Variable, 130, 333, 377
bound, 166
capture, 175
class, 285
external, 183
global
static allocation, 93
in activation record, 95
in functional languages, 333
in imperative languages, 130
in logic programming, 377
instance, 284
logic, 377
modifiable, 130
reference model, 131
shadowing, 289
static, 168
Variance, 381
Variant, 211
in C, 212
in Pascal, 211
security, 213
Visibility, 73
Visibility rule, 67
Void, 207

Von Neumann, 334, 413, 414
Vtable, 303

**W**
Warren Abstract Machine, 408
Wilkes, 413
Wirth, 418

**X**
XML, 428