INDEX

Symbols

(vector), 18
#f (false), 5, 27
#false (false), 27
#t (true), 5, 27
#true (true), 27
λ (lambda), 42
ϕ (phi, golden ratio), 67
’ (quote), 3
* (multiplication operator), 29, 30
+ (addition operator), 29, 30
- (subtraction, negative operator), 29, 30
/ (division operator), 29, 30
< (numeric less than), 33
<= (numeric less than or equal), 33
= (numeric equality), 12, 33
>= (numeric greater than or equal), 33
15 puzzle, 198–215

A

A* search algorithm, 200–204
abs function, 36
abstract syntax tree, 272
abstract syntax tree (AST), 276
accessor method (structs), 21
addition operator (+), 29, 30
admissible, 201
alphametics, 245
and function, 28
animation, 119–124
append function, 7
apply function, 43
arity, 310
assigning, 9
AST (abstract syntax tree), 276
atan function, 36
atomic data, 1
automated testing, 309

B

backtracking, 184, 220, 232
bag, 96
Ball, W. W. Rouse, 64
BFS (breadth-first search), 182
Big O notation, 191
binary heap, 191
binary numbers, 30
binary search, 156–158
binary tree, 4
bind, 9
Binet’s Formula, 71
bitmap-dc% object, 92
body, 232
Boles and Creots, 254
Boolean algebra, 27
Boolean functions
  and, 28
  not, 28
  or, 29
Booleans, 27–29
bound identifiers, 2
bounding box, 94
brackets
  curly, 3
  square, 3
breadth-first search (BFS), 182
brush object, 93
brush% object, 93
Bulls and Cows, 254
button% object, 118

C

call-with-input-file expression, 147
call-with-output-file expression, 147
canvas% object, 119
capture, 58
car, 4
car function, 5
Cartesian product, 104
cdr, 4
cdr function, 5

`ceiling` function, 36
CFG, 288
character class, 277
characters, 12–13
child node, 181
circuit, 180
clauses, 228
Clocksin, William F. and Mellish, Christopher S. *Programming in Prolog*, 234
`close-input-port` expression, 145
`close-output-port` expression, 145
closure, 58
closures, 58–60
coefficient of determination, 175
comparison operators, 33–34
complement, 99
complex numbers, 32–33
complexity, 191
`compose` function, 44
composed, 44
compound interest, 165–166
computer ports, 148
`cond` expression, 47
conditional expressions, 47–48
connected graph, 180
`cons` function, 5
cons cell, 4
context-free grammars, 288
continued fraction, 71
corporation, 149
`cos` function, 36
covariance, 174
cryptarithm, 245
cryptarithmetic, 245
CSV files, 149
curvature, 68
cut, 244
cycle, 180
cycloid, 81

D

data files, 150–155
data visualization, 159–171
database, 155–158, 228
DC, 92
`define` (function definition), 42
`define` (identifier assignment), 9
`denominator` function, 31
dependent response, 173
depth-first search (DFS), 182, 247
deterministic FSM, 260
DFA, 260
DFS (depth-first search), 182
digit, 317
Dijkstra's shortest path algorithm, 189–198
Dijkstra, Edsger, 189
directed acyclic graph (DAG), 180
`discrete-histogram` function, 168
`display` expression, 25
division operator (`/`), 29, 30
`do` form, 51
do statement, 51
drawing, 92–115
drawing context, 92
Dudeney, Henry, 245

E

EBNF (extended Backus–Naur form), 288
edge, 180
elementary data types, 1
encapsulate, 62
`eof-object?` function, 146
`eq?` function, 11
`equal?` function, 11
equality, 11–12
erasing drawing context, 99
event, 117
event table, 261
exact numbers, 29
`exact?` function, 33
exception handling, 301
exclusive-or, 112
executable files, 314–315
`exp` function, 36
expression, 9
`expt` function, 36
extended Backus–Naur form (EBNF), 288

F

false (`#f`, `#false`), 5, 27
Fibonacci sequence, 66–72, 105–109
field, 21
FIFO, 60

file I/O, 145–147
filter function, 45
finite state automaton (FSA), 259–262
finite state machine (FSM), 259–262
first function, 5
fixnums, 11
floating point numbers, 31–32
floor function, 36
foldl function, 44
foldr function, 44
for forms, 52–57
for form, 53
for* form, 54
for/and form, 56
for/first form, 57
for/last form, 57
for/list form, 55
for/or form, 56
for/sum form, 57
form, 9
fractions, 30
FSA (finite state automaton), 259–262
FSM (finite state machine), 259–262
function, 8, 41
function expression, 77
Functional programming, 43

G

get-output-string function, 147
Ghostbusters, 34
global environment, 45
goal, 228
golden ratio, 67
golden ratio (ϕ), 67
golden spiral, 67
gradient, 138
graph, 180
graph theory, 180–182
graphical user interface, 117
GUI, 117

H

Hamiltonian circuit, 181
Hamming distance, 202
hash set, 96
hash table, 59, 70
hash-has-key? function, 59
hash-ref function, 59
hash-set! expression, 59
head, 231
heuristic, 179, 200
hexadecimal, 30
higher order functions, 43–45
higher-order functions, 43
Hindu–Arabic numeral system, 317
histogram, 167
hypocycloid, 83

I

identifiers, 9, 10
if expression, 47
imag-part function, 33
immutable, 19
imperative programming, 51
impure functions, 51–57
in-naturals function, 52
in-range function, 52
independent predictor, 173
index fund, 149
index-of function, 8
inexact, 31
infix notation, 37–39
Insurance Salesman problem, 72–74
Integers, 29
integers, 29–30
interpreter, 276
inversion, 199
iteration, 48–51

K

keyword, 10
keyword->string function, 17
Kleene plus, 279
Kleene star, 279

L

lambda (λ), 42
lambda expression, 42
lazy evaluation, 52
leaf node, 181
length function, 6
let (named), 50
let expression, 46
let* expression, 46
lexical analyzer, 276–288
lexical scoping, 45–47

linear algebra, 126
linear regression, 173
linear-gradient% object, 138
Lisp, 2
list form, 2
list comprehension, 55
list functions
  append, 7
  car, 5
  cdr, 5
  cons, 5
  first, 5
  index-of, 8
  length, 6
  list-ref, 6
  list?, 5
  make-list, 7
  member, 8
  null?, 7
  pair?, 5
  range, 7
  rest, 5
  reverse, 6
  second, 6
  sort, 7
  third, 6
list->string function, 17
list->vector function, 20
list-ref function, 6
list? function, 5
lists, 1–8
  list form, 2
  quote form, 3
  structure, 4–6
LL(1) grammar, 291
log function, 36
logic programming, 227–258
loops, 48–51
Loyd, Sam, 198

M

make-bitmap function, 92
make-color function, 93
make-gradient function, 138
make-hash function, 59
make-list function, 7
make-object function, 93
make-string function, 14
make-vector function, 20
Manhattan distance, 201
map function, 43
market index, 149
Mastermind, 254
max-heap, 192
mcar function, 61
mcdr function, 61
mcons function, 61
member, 96
member function, 8
memoization, 59
message% object, 118
min-heap, 192
multiplication operator (*), 29, 30
multiset, 96
mutability, 48
mutable, 19
mutable cons cell, 61
mutate, 9
mutators, 22

N

n-Queens Problem, 183–188
negative operator (-), 29, 30
NFA, 260
nil, 4
Nim, 109–115
nim-sum, 112
node, 180
nondeterministic FSM, 260
nonterminal, 289
not function, 28
not equals, 34
null? function, 7
numerator function, 31
numeric comparison
  equality (=), 12, 33
  greater than or equal (>=), 33
  less than (<), 33
  less than or equal (<=), 33
numeric functions
  abs, 36
  atan, 36
  ceiling, 36
  cos, 36
  exp, 36
  expt, 36
  floor, 36
  log, 36

sqr, 36
sqrt, 36
tan, 36
numeric type tests, 35–36
numerical tower, 29

O

object-oriented programming, 60
octal, 30
open-input-file function, 145
open-output-file function, 145
open-output-string function, 147
or function, 29
output control, 24
output expressions
display, 25
print, 25
printf, 26
println, 26
write, 24

P

pair, 4
pair? function, 5
parameter, 79
parametric plots, 79–86
parser, 276, 288–298
path, 180
paths, 100
Pattern matching, 151
pattern matching, 271
PDA (pushdown automaton), 267–272
pen object, 92
pen% object, 93
permutations, 242–245
phi (ϕ, golden ratio), 67
Pigs and Bulls, 254
plot function, 76
plotting, 75–91
plotting points, 86–88
point, 86
polar plots, 88–91
port functions
call-with-input-file, 147
call-with-output-file, 147
close-input-port, 145
close-output-port, 145
eof-object?, 146
get-output-string, 147
open-input-file, 145
open-output-file, 145
open-output-string, 147
read-line, 145
tcp-accept, 148
tcp-listen, 148
ports, 145
predicate, 11, 228
prediction line, 173
predictor variable, 173
print expression, 25
print precision, 18
printf expression, 26
println expression, 26
priority queue, 189, 191–194
production rule, 288
Programming in Prolog (Clocksin and Mellish), 234
proper subset, 104
pure functions, 48
pushdown automaton (PDA), 267–272
pushdown stack, 267

Q

quantifiers, 279
query, 228
queue, 60
quote form, 3
quote ('), 3

R

Racketeers, xx
Racklog, 227
Racklog comparison operators
%<, 236
%=/=, 236
%=:=, 236
%>=, 236
Racklog forms
%append, 238
%assert!, 231
%assert-after!, 231
%bag-of, 233
%fail, 237
%member, 240
%more, 229
%rel, 228

Racklog forms, *continued*
%true, 237
%which, 228
Racklog logic operators
%and, 237
%not, 237
%or, 237
Racklog predicates
%/=, 235
%/==, 235
%=, 234
%==, 235
%if-then-else, 237
%is, 235
%let, 235
Random file access, 155
range function, 7
Rational numbers, 29
rational numbers, 30–31
read-line function, 145
real numbers, 29, 31–32
real-part function, 33
recognizers, 260
recursion, 49
recursive descent parsers, 291
regexp values, 280
regexp-match function, 280
regexp-match* function, 283
regexp-match-positions function, 280
regression, 173–178
regression analysis, 173
regression line, 173
regular expressions, 279–286
regular language, 272
renderer, 77
REPL, 255, 256
rest function, 5
rest-id, 299
reverse function, 6
river crossing puzzles, 246
rule, 231

S

s-expression, 3
second function, 6
securities, 149–150
send expression, 64
sequence expressions, 54
server, 148
set, 96
set difference, 102
set intersection, 102
set theory, 96–105
set union, 99
set! expression, 9
set-builder, 99
set-mcar! expression, 61
set-mcdr! expression, 61
side effects, 43, 48
sort function, 7
split, 150
sqr function, 36
sqrt function, 36
squared correlation, 175
standard deviation, 171–173
state table, 260, 264
state-transitions diagram, 260
statistics, 171–178
stream, 52
string, 13
string function, 14
string conversion and formatting, 17–18
string functions
keyword->string, 17
list->string, 17
make-string, 14
string, 14
string->list, 17
string->symbol, 17
string-append, 13
string-contains, 16
string-copy, 14
string-downcase, 15
string-keyword, 17
string-length, 15
string-ref, 14
string-replace, 16
string-set!, 14
string-split, 16
string-titlecase, 15
string-trim, 16
string-upcase, 15
string<=?, 15
string=?, 15
substring, 15
symbol->string, 17
string ports, 147–148

string->keyword function, 17
string->list function, 17
string->symbol function, 17
string-append function, 13
string-contains? function, 16
string-copy function, 14
string-downcase function, 15
string-length function, 15
string-ref function, 14
string-replace function, 16
string-set! expression, 14
string-split function, 16
string-titlecase function, 15
string-trim function, 16
string-upcase function, 15
string<=? function, 15
string=? function, 15
strings, 13–18
struct form, 21
structs, 21–24
  accessor method, 21
subsets, 103
substring function, 15
subtraction operator (-), 29, 30
Sudoku, 215–225
sum of squares errors, 173
symbol, 10
symbol->string function, 17
symbolic expressions, 3
symmetric difference, 103
syntax diagram, 277

T

table->string function, 152
tail call optimization, 50
tan function, 36
tcp-accept function, 148
tcp-listen function, 148
TCP/IP, 148
terminal node, 181
terminals, 288
text-table package, 152
textual environment, 45
third function, 6
timer object, 120
timer% object, 119
Tower of Hanoi, 64, 133
TRAC, 275
transformations, 98
transposition, 199
tree, 180
true (#t, #true), 5, 27
Turing complete, 263
Turing machine, 263–267

U

unbound identifiers, 2
Unicode, 12
unification, 229
unless expression, 48

V

variable, 9
variance, 172
vector function, 18
vector functions
  list->vector, 20
  make-vector, 20
  vector, 18
  vector-append, 20
  vector-immutable, 19
  vector-length, 20
  vector-list, 20
  vector-member, 21
  vector-ref, 19
  vector-set!, 19
  vector-sort, 20
vector, quoted (#), 18
vector-append function, 20
vector-immutable function, 19
vector-length function, 20
vector-list function, 20
vector-member function, 21
vector-ref function, 19
vector-set! expression, 19
vector-sort function, 20
vectors, 18–21
Venn diagrams, 97
vertex, 180

W

walk, 180
weight, 181
weighted graph, 181
when expression, 48
word addition, 245
write expression, 24

X

x-y plots, 76–78

Z

zero-based indexes, 6