《高阶 Perl》索引[1]

Special Characters

"" (stringification) pseudo-operator, 312
"" key, 334
$" special variable, 174, 204
$$ special variable, 101
$. special variable, 235
$/ special variable, 174, 240
$; special variable, 55
$;$ prototype, 225
$@ special variable, 287
$_ special variable, 93, 104
& notation in Linogram, 330
&@ prototype, 225
&;@ prototype, 225
&& @rest condition, 139
(?!) regex lookahead operator, 242
* operator, 181, 259, 372
* quantifier, 291
* token, 251
** operator, 284
+ operator, 372
+ quantifier, 291
+ token, 251
-> operator, 116
. operator, 116
; symbol in prototypes, 118
<...> operator, 123, 239–243
  as rudimentary parser, 239
<=> operator, 67, 71
== operator, 111, 117
  for comparing objects, 319
? quantifier, 291
@_, modification of, 59–60
[...] operator, 110
\ (backslash) operator, 55, 57, 118, 293, 300
\@$$ prototype, 195
\@\@ prototype, 117
| operator, 291
||= operator, 70
|| operator, 67

A

-a command-line option, 98
$a and $b variables, 230
Abelson, Harold, x
absolute zero 绝对零度, 317
abstract base class 抽象基类, 17
abstract syntax tree (AST) 抽象语法树, 37–38, 263, 267, 272, 292, 369–371
abstraction of functionality 功能性提取, 11, 27, 149, 275
accessor method 访问器方法, 19, 319
actions 行为, 29, 32–33, 35, 297
Adler, David, 95
agenda 安排表, 137–138, 243, 247, 254
agenda method, universality of 安排表方法, 144
aggregation 合计, 13
airplane 飞机, 72
algorithm 算法, 7, 8, 191, 315
aliasing 别名, 53, 60
alternate universe 另一个世界, 75–76
alternation of parsers 交替, 259
ambiguity in arithmetic expressions 数学表达式的歧义, 35
amortization 分期还款, 207–209
anchor component of URL; URL 的锚组成部分, 125
anonymous array 匿名数组, 21, 32, 53
anonymous functions 匿名函数, 13, 50, 52, 56, 80, 171–173, 200, 218, 247, 280, 302, 319
  debugger's treatment of 除错器的处理, 277
argument isn't numeric warning, 71
argument normalizer, inlined cache manager with 参数归一化，内联的缓存管理, 59–60
arithmetic expressions 算术表达式, 35, 251, 261–290
  overview 概况, 261–267
  calculator 计算器, 267, 283–285
  debugging 除错, 277–282
  generic-operator parsers 通用操作符解析器, 275–277
  grammar for 语法, 250
  left recursive 左递归, 267–272
arrays 数组, 43, 118, 119
  contrasted with iterators 与迭代器对比, 76–77
  contrasted with linked lists 与链表对比, 171
  looping over 循环, 117–120
  representation of database rows as 数据库列的表示, 108–109
  repeated copying of 重复复制, 84
arrow, definition in Linogram, 333
Ashton, Elaine, 95
assembly language 汇编语言, 210
associativity of operators 操作符的结合性, 268, 284
AST (abstract syntax tree) 抽象语法树, 37–38, 263, 267, 272, 292, 369–371
asymmetry, in Linogram parameter specifications, 373
atomic expression 原子表达式, 291
atomic features in Linogram, 330
atomic ray, 122
atoms in regexes 正则中的原子, 291, 293
attribute-value pairs in HTML tags 在 HTML 标签中的属性－值对, 17
automatic profiling 自动剖析, 73–74
auxiliary parameter 辅助的形参, 160

B

\b regex metacharacter, 181, 293
backslash character 反斜杠字符, 55, 57, 118, 293, 300
backtracking 回溯, 136, 304–310
  in regex engine 在正则引擎中, 242
backtracking parsers 回溯解析器, 304–310
  overview 概况, 304
  continuations 续篇, 305–308
  parse streams 解析流, 308–310
backup program 备份程序, 136
bacteriophage 噬菌体, 88
bad link detector 坏链接的检测器, 126–128
bad luck 不幸, 111
balanced parentheses 平衡的括号, 191–192
barbarian 原始人, 136–137
base case 基本型, 2, 3, 6, 139, 356
bear, 239
BEGIN block, 225–226, 230, 233–234, 236–238, 336
beginner mode via alternative dispatch table 通过可选的分配表的初学者模式, 32
behavior function in constraint system 约束系统中的行为函数, 320, 321
Bell Labs, 375
BFS (breadth-first search) 宽度优先搜索, 125, 132, 140–142
binary 二进制, 1–2, 159–160
binary search 二分搜索, 195
binding 绑定, 47
bio-informatics 生物信息学, 88–91
Bird, Richard, x
blessing, 238, 277, 311, 313
blocks 块, 47–48, 260
  bare 空的, 44

[1]本索引由译者根据原书索引制作。最新版可下载自：http://hop.perl.plover.com/chinese.html

as arguments to **sort()** function 作为 **sort()**函数的参数, 67
BEGIN, *see* **BEGIN** block
disk, *see* disk blocks
labelled, parser for, 361
boolean “not”operator, 236
Boss, the 老板, 53
box, definition in Linogram, 332
breadth-first search (BFS) 宽度优先搜索, 125, 132, 140–142
browsers 浏览器, 44
buffer 缓冲区, 241
bullet symbol 着重号, 294
Burke, Sean, x, xi, 19
bytes 字节, 13–14

C

C programming language; C 编程语言, ix, 44, 49, 52, 69, 71, 100, 109, 230, 241, 243, 283, 335
auto variables in 自动变量, 4
for loop, 114
library of 库, 100, 101
C++ programming language; C++ 编程语言, 17
/c regex modifier flag, 244
cache data, expiration of 缓存数据的到期, 64
cache hit 缓存命中, 42, 59, 68, 70
false 错误的, 55, 70
cache management overhead 缓存管理的开销, 42, 57, 66, 71
cache memory 缓存内存, 44, 71
cache miss 缓存脱靶, 42, 59, 62, 70
caching and memoization 缓存和记忆术, 41
alternatives to 可供选择的, 66–71
benefits of speed 速度的好处, 71–74
caveats 警告, 52–55
evangelism 传播福音, 71
inline caching 内联缓存, 43–44
in object methods 在对象方法里, 62–64
key generation 键的生成, 55–62
Memoize module, 45–52
persistent caches 持续的缓存, 65–66
quantitative analysis of 定量的分析, 42, 54–55, 59, 66
and recursion 和递归, 42–43
calculator 计算器, 35–40, 267, 283–285, 290, 313
calculus 微积分, 201, 204, 210, 213
call frame 调用框架, 153–154
call tree 调用树, 142–143, 146
callbacks 回调, 7, 13–16, 78, 83–84, 93–94, 103, 105, 119–120, 124–126, 149, 151, 218, 247, 258
caller() function, 280
canonical form 规定形式, 144–145
canonical order 正规的次序, 233
cantankerous grandfather 坏脾气的爷爷, 151
cardioid 心脏线, 327
Carp module, 122, 348, 355, 372
carriage return 回车, 293
carrot, 106, 175, 184
CGI, 125, 130
CGI::Push module, 102
chained lexers 链式词法分析器, 245–249
character classes 字符类, 182, 187
CHDIR directive in configuration file, 27
chdir() function, 27
Christiansen, Tom, ix, 84
chromosome 染色体, 88
chronological order 时间次序, 67–69, 98
CIA, 122
circle, definition in Linogram, 333
clauses in grammars 语法中的条款, 251
cleverness 才智, 139
clock, memoized 带记忆的时钟, 54
clone method in **URI::URL**, 132
closedir() function, 10
closure 闭包, 50, 80, 217
closure property 闭包属性, 49–51
closure operator (*) 闭包算符, 181, 191
cmp operator, 69
CMYK, 41
code reference 代码引用, 6–7, 27, 58, 104, *see also* cosure, anonymous function
code value (CV), 49–50
coefficient 系数, 211, 334, 339
college student, penurious 穷大学生, 136
comma-separated list 逗号分割的列表, 361, 370
comparator function 比较器函数, 66–67, 194–195, 233
compiler 编译器, 37
comp.lang.perl.misc newsgroup, 181
components 组成部分, 18
compound interest 复利, 207–209
compromise 折中, 56
compulsive behavior 难以抑制的行为, 303
concatenate() operator, 265–266
concatenation 组合/连接, 2, 181–182, 258
of parsers, 258
of regexes, 181–182
configuration table 配置表, 27, 30
CONSTANT object, 343–345
constant part of equation 方程的常数部分, 334–336
constants 常数, 323
constraint networks 约束网络, 326
constraint systems 约束系统, 315
Constraint class, 342
constraints section in Linogram specification, 366–367
Constraint_Set class, 343
_content key in **HTML::TreeBuilder**, 18
context 上下文, 68, 77, 104, 108, 120–121, 126, 128
Linogram evaluation environment, 371–373
continuation 续篇, 305–308
control-backslash character 控制 —反斜杠字符, 55
conversion, decimal to binary 转换，十进制到二进制, 1–2
core dump, 49
cos() function, 209–210, 213
cosh() function, 213
cost-benefit ratio 付出 —回报比, 72
counting process 数数过程, 85–86
CPAN, 69, 99, 216, 334
CPU, 44
crab cakes 蟹肉蛋糕, 54
current matching position of scalar 标量的当前匹配位置, 244–245
Curry, Haskell B., 218
currying, 217–221, 227–229, 231, 234, 236, 323
custom key generation 定制的键生成, 62
cutsorting 切断排序, 192–200
log files 日志文件, 195–200
cutting function 切断函数, 194
CV (code value), 49–50
cwd() function, 30
Cwd module, 30

D

-d operator, 81
dangling symbolic link 悬空的符号链接, 13, 16, 121, 217
database handle 数据库句柄, 92

database query languages 数据库查询语言, 315
database-query parsing 数据库查询解析, 299–304
  overview 概况, 299
  lexer 词法分析器, 299–301
  parser 解析器, 301–304
databases 数据库, 234–238, 299–304
  of cached values 缓存值的, 65–66, 73
  flat 纯文本, 91–100
  SQL, 107–111
Data::Dumper module, 262
data structure 数据结构, 55, 362
date format, ISO 日期格式，ISO, 69
DB_File module, 65
DBI module, 107–110, 114
DBM, 66, 233
dead ends 死胡同, 125
debugger 除错器, 101–102
debugging 除错, 101–102, 277–278, 286, 317, 320, 367
decimal to binary conversion 十进制到二进制转换, 1–2
declarative programming 声明式编程, 315–375
  overview 概况, 315
  linear equations 线性方程, 326–327
  local propagation networks 本地传播网络, 315–326
    overview 概况, 315–317
    implementing 实现, 317–325
    problems with 问题, 325–326
declarator in Linogram, 364, 365
default action for dispatch table 分配表的默认行为, 34–35
Defenestration of Prague 布拉格的扔出窗外事件, 134
defined operator, 110
depth-first search, *see* DFS
derivation of sentence 句子的推导, 251
derivative function 导函数, 203–204, 213, 214, 220
DESTROY method, 63
destructive function 破坏性的函数, 59, 174, 197
destructive method 破坏性的方法, 338
DFS 深度优先搜索, 124, 132, 134, 140–142, 151–152, 255, 269
  ambiguity of 歧义, 152
  bound on size of agenda 待办清单的尺寸限制, 140
  contraindications for 禁忌, 140–141
  recursion and 递归, 82, 124, 134, 141, 143
  web spiders and 网页抓取, 141
diagnostic messages 诊断消息, 27, 133, 173, 318
diamond, definition in Linogram, 334
die(), throwing exceptions with, 287
differential calculus 微分学, 203, 213
Dijkstra, Edsger Wybe, 145
directives 指示, 29–30
directory tree 目录树, 80, 106
directory walking, applications and variations of 目录遍历，应用和变化, 11–16
dirhandles 目录句柄, 9–10, 77
disk blocks 磁盘块, 10
diskettes 软盘, 136
dispatch tables 分配表, 27–40, 293, 343–345, 347, 352, 368
  overview 概况, 27
  advantages of 优势, 29–32
  calculator 计算器, 35–40
  configuration file handling 配置文件处理, 27–35
  default actions 默认行为, 34–35
division as inverse of multiplication 除法是乘法的逆运算, 323, 343–345
division by zero 除以零, 288, 322, 345
division of power series 幂级数的商, 214–216
division of treasure, *see* partition problem
do-it-yourself, 82
do-while loop, 162
domain name server 域名服务器, 44, 71
doorbell 门铃, 258
drand48() function, 100–101
draw section in Linogram specification, 367–369
drawables 可绘制的, 358–359, 367, 369
drawables list 可绘制列表, 369
drawing function 绘制函数, 330–331, 358, 374
drunken fraternity brothers 喝醉的好兄弟, 102
du command in Unix, 11
duration (of value) （值的）有效期, 47–49

E

-e command line option, 98
-e operator, 13
each() operator, 120, 233
$elementfunc argument, 20, 38–40
email 电子邮件, 195–196, 241–242
  address 地址, 35, 196
  header of 头部, 241–242
empty list 空列表, 16, 24, 109, 118, 231–232, 257
empty string 空字符串, 182, 190, 248, 292, 341
end tag in HTML; HTML 中的结束标签, 17
Engineering Mathematics Handbook, 216
entropy 无序, 102
environment 环境, 50
Environment class, 359–360
epoch format 纪元格式, 197
eq operator, 111, 302
Equation class, 334–341
Equation::System class, 339–342
equations,
  for Fibonacci numbers, 23
  inconsistent 前后矛盾的, 340
  linear 线性的, 326–343
  redundant 多余的, 339
  roots of 根, 201
  solving 解, 200
  trivial (0=0), 336
equivalence classes 等同类, 145
error diagnosis and recovery in parsing 解析中的错误诊断与矫正, 285–290
escape sequence 转义序列, 55
eta-conversion, 292
Euclid’s algorithm, 153
eval operator, 30, 56–58, 226, 227, 287, 362
exaggeration 夸张, 23
exceptions 例外, 287–290
  propagation of 传播, 288
execute method in **DBI** module, 110
EXHAUSTED operator, 113–114
exhaustion 耗尽, 107, 124, 137–138, 195, 233, 243, 247, 286, 306
  explicit function for 明确的函数, 112–114
  synthetic representation of 人为的表示, 110–114
exists() operator, 70
exp() function, 213
expiration of cache data 缓存数据的到期, 64
explicit exhaustion function 明确的耗尽函数, 112–114
explicit specification of empty list value, 231
explicit stacks 明确的栈, 161–169
exponentiation operation 求幂操作, 283

Expression class, 369
expression parser 表达式解析器, 36–37, 250, 254–256, 261–277
expressions 表达式, 251–253
  in Linogram, 369–373

F

-f operator, 9
@F special variable, 98
factor of arithmetic expression 算术表达式的因子, 263
factorial function 阶乘函数, 2–4, 160, 209–210, 216
fame and renown 声誉与名望, 277
family tree 家族树, 327
Fcntl module, 93
features in Linogram, 327–332, 347, 352–358
Fenchurch, 54
FETCH method, 121–122
fetchrow_arrayref method in **DBI** module, 108, 110, 114
Fibonacci numbers, 22–23, 42, 43, 162–169
  eliminating recursion from calculation of 从计算中移除递归, 162–169
fields in email message header 电子邮件消息头部的段落, 242
fields in flat text database 纯文本数据库的字段, 91, 97, 98
File::Basename module, 126
File::Find module, 82
File::ReadBackwards module, 99
file system 文件系统, 8, 17
filehandle 文件句柄, 75–76, 123
  as iterator 作为迭代器, 75–76
  iterator interface to 迭代器界面, 91
  reading in **while** loop 在**while**循环中读取, 123
  tied 系住的, 123–124, 199
files 文件, 8–9, 12, 16, 54
  configuration 配置, 27–34
  for persistent caching 持续的缓存, 65–66
  GIF, 41
  INI, 76
  log 日志, 97–100, 195–200
  with octopuses 含有章鱼, 106
  with plutonium 含有钚, 75–76
filters and transforms 过滤和变换, 103–106
fishing expedition 钓鱼, 72
flat databases 纯文本数据库, 97, 302
FlatDB module, 91–100, 238, 299
FlatDB::Iterator module, 98–100
flat-file database 纯文本文件数据库, 91–97
floating-point numbers 浮点数, 204, 205
flow chart 流程图, 103
foolishness 愚蠢, 52
forcing the promise 推动断言, 172, 179
for loop, C-style, 114
foreach, using to loop over more than one array, 117–120
fork() function, 53
Fortran, 4
four-dimensional beings 四维生物, 346
fractions 分式, 153
fragment of URL, *see* anchor component
FreezeThaw module, 55
Frege, Gottlob, 218
Frequently Asked Question, 1, 11, 84
fruit 水果, 156
function calls, implementation of 函数调用的实现, 152
function factory 函数工厂, 45–46, 131, 172, 217, 257, 271
functional vs. object-oriented programming 函数式编程 vs. 面向对象式编程, 17
functions,
  custom key generation, for impure functions 为非纯函数定制的键生成, 62
  higher-order 高阶, 217, 222
  that return references 返回引用的, 53–54
  very fast functions 非常快的函数, 54–55
  whose return values do not depend on their arguments 返回值不依赖于参数, 52–53
  with reference arguments 带有引用参数的, 61
  with side effects 有边界效应的, 53

G

\G regex metacharacter, 244
/g regex modifier flag, 244
garbage collection 垃圾回收, 47, 50, 52
Gaussian elimination 高斯消元法, 334, 336–342
GCD (Greatest Common Divisor) 最大公约数, 153
geeks 爱好者, 207
General Electric 通用电气, 45
generic-operator parsers 通用操作符解析器, 275–277
Genomic sequence generator 基因组序列生成器, 88–91
getc() function, 123
get_links method of **HTML::LinkExtor** module, 125–126
GIF file, 41
globs, 123, 220
glob() operator, 77
global filehandle 全局文件句柄, 31
global variables 全局变量, 3–4, 30, 32, 93, 176, 230, 280, 303
  effect on recursion 于递归的效果, 3–4, 9–10
  filehandles as 作为文件句柄, 9–10
  hashes instead of 替以散列, 33
  random number generator as 作为随几数生成器, 102
Goff, Jeff, 144
golden ratio 黄金比例, 23
golden_rectangle, definition in Linogram, 333
goto, 154, 162, 311
grammar 语法, 250–254
Grasshopper, 124–133
greatest common divisor (GCD) 最大公约数, 153–155
grep() function, 222–223
grocery bags 杂货袋, 24

H

hack, 55
hairy code 可怕的代码, 276
Hall, Joseph, 70
Hamming problem, 179–181
Hamming, Richard W., 179
hash 散列, 13–16, 18, 26, 33, 43, 55, 58–59, 73, 132, 156, 197, 320, 330–331, 341, 346, 374
  cache 缓存, 41–43, 45–46, 62–63, 66
  as dispatch table 作为分配表, 29
  for canonicalizing value 为了规范值, 145
  iterator component of 迭代器成分, 77, 120
  power set of 幂集合, 156–157
  tied 系在/关联, 66, 277
hash key 散列键, 55, 61, 63
hash slice 散列的片段, 94
hash value computation 散列值的计算, 55
head of linked list 链表的头部, 170
header of email message 电子邮件消息的头部, 241

header of flat-file database 纯文本文件数据库的头部, 91
headers of HTML document; HTML 文档的头部, 20
hexadecimal 十六进制, 63
hierarchical data 层次化数据, 8–11
Hietaniemi, Jarkko, 156
high school 高中, 200, 208, 327
higher-order functions 高阶函数, 217, 222
  currying, 217–238
  reduce() and **combine()**, 229–234
hline, definition in Linogram, 331
Hoefler, Jonathan, 134
Hoefler Type Foundry; Hoefler 铅字厂, 134
hook 钩子, 112, 124, 133, *see also* cllback
HTML, 17–22, 38–40, 118, 124–125, 217–218
HTML::LinkExtor module, 124, 125
HTML::TreeBuilder module, 18–19
HTTP, 98
httpd log file, 97–98
Human Genome Project 人类基因组项目, 88
hyperlinks 超链接, 124–130, 132, 141

I

if-else tree, 27–28, 37, 39, 163
impure function 非纯函数, 62
INCLUDE directive in configuration file, 32
incrementing numerals 增大数字, 86
India 印度, 199
infinite loop 无限循环, 76, 205, 269, 318, 338
infinite sequence of integers 无限的整数序列, 173
infinite streams 无限流, 170–216
  overview 概况, 170
  hamming problem, 179–181
  lazy linked lists 惰性链表, 171–175
  Newton-Raphson method, 200–209
  power series 幂级数, 209–216
  recursive streams 递归流, 175–179
  regex string generation 正则字符串生成, 181–200
infix form of expression 表达式的中缀式, 36
ingenuity 独创性, 44
INI file, 76
inline caching 内联缓存, 43–44
inlined cache manager, with argument normalizer 内联的参数归一化的缓存管理, 59–60
inlining 内联, 57–60, 70
internal stack 内部的栈, 82
intrinsic constraints 内在的约束, 347–348
Intrinsic_Constraint_Set class, 348
Introduction to Functional Programming, x
i-number, 196
investment banking 投资业, 62–64, 70
IOU, 173
IRC, 11, 181
is_in() example, 61
isa method, 57
ISO date format; ISO 日期格式, 69
iteration, C model of; C 的迭代模型, 114
iterators 迭代器, 75–133
  overview 概况, 75
  alternative interfaces to 可选的界面, 116–124
  converting recursive functions to 转换递归函数到, 134–169
  examples 实例, 82–103
  exhaustion of 耗尽, 79
  filehandle iterators 文件句柄迭代器, 91
  filters and transforms of 过滤和变换, 103–106
  flat-file database 纯文本文件数据库, 91–97
  genomic sequence generator 基因组序列生成器, 88–91
  homemade 自制, 78–82
  introduction 介绍, 75–77
  kicking 踢, 79, 103, 105, 122, 129
  for permutations 排列, 83–88
  random number generators 随机数生成器, 100–103
  rewinding 倒回, 170
  searching databases backwards 反向搜索数据库, 97–100
  semipredicate problem 半谓词问题, 107–116
  web spiders 网页抓取, 124–133

J

jargon 行话, 251
Junko, 54

K

KEEPER label for HTML elements, 20–21, 217
key generation 键的生成, 55–62
  overview 概况, 55–58
  functions with reference arguments 带有引用参数的函数, 61
  partitioning 划分, 61–62
key-generation method 键生成方法, 56
keys() function, 77
kicking an iterator 踢迭代器, 79, 103, 105, 122, 129
Kleene closure, 181
Knuth, Donald E., 101, 206

L

-l operator, 13
labeled block 贴标签的块, 361
Latin 拉丁文, 52, 137
laws of programming 编程的法则, 28
lazy linked lists 惰性链表, 171–175
left-associative operator 左结合操作符, 284
left recursion 左递归, 267–273
left-recursive grammar rules 左递归语法规则, 269
Leler, Wm, xi
lemniscate 双扭线, 327
Leonardo of Pisa (Fibonacci), 22
lexer, Perl's; Perl 的词法分析器, 301
lexers 词法分析器, 243–250, 295, 299–301, 360–361
  overview 概况, 239–240
  chained 链式的, 245–249
  emulating **<>** operator 评估 <> 操作符, 240–243
  peeking 偷看, 249–250
lexical analysis, *see* lexers
lexical closure 词法闭包, 49
lexical filehandle 词法文件句柄, 31
lexical variables 词法变量, 44, 60, 279
lexicality, 47
lexing, *see* lexers
line, definition in Linogram, 330
linear equations 线性方程, 326–327
linked lists 链表, 170–171
  head 头部, 170
  lazy 惰性, 171–175
  splicing into 拼接, 171
  tail 尾部, 170
linogram, 327–334
  overview 概况, 327–334
  equations 方程, 334–343
  feature types 特征类型, 354–360
  missing features 缺失的特征, 373–375
  parser 解析器, 360–373
    overview 概况, 360–361
    declarations 声明, 363–369
    definitions 定义, 363–363
    expressions 表达式, 369–373

required extensions 需要的扩展, 361–362
programs 程序, 362–362
scalar types 标量类型, 354–355
%TYPES, 362
values 值, 343–354
overview 概况, 343–345
constant values 常数值, 345–345
feature values 特征值, 347–347
feature-value methods 特征值方法, 352–354
intrinsic constraints 内在的约束, 348–348
synthetic constraints 综合的约束, 348–352
tuple values 元组值, 345–347
Lisp programming language; Lisp 编程语言, ix
list assignment as condition 条件里的列表赋值, 109
list context 列表上下文, 68, 77, 104, 108, 120, 126, 128
list expressions 列表表达式, 260
List::Util module, 229–232
literal regexes 字面正则, 182
local() operator, 93, 105, 204
local propagation networks 本地传播网络, 315–326
overview 概况, 315–317
implementing 实现, 317–325
problems with 问题, 325–326
log files 日志文件, 97–100, 195–200, 232–233
log() function, 213
LOGFILE directive in configuration file, 27
lookahead assertions 前瞻断言, 293
lookbehind assertions 后顾断言, 293
loop control operators 循环控制操作符, 162
love and admiration 喜爱并钦佩, 11
ls command in Unix, 11
ls -R command in Unix, 11
Lucas, Edouard, 4
Luke the Hermit, 70
LWP::Simple module, 124–125, 133

M

-M operator, 54
m//g operator, 77
Macdonald, John, 156
Macintosh, 136
magic variable 魔幻变量, 121, 123
mailto URL, 132
maintenance programmers 维护程序员, 57
make_counter example, 50–52
malloc() function, 109
map() function, 81, 103–104, 174, 222–224, 230, 265, 348, 349
Marshall, Edward Waite, 45
marshalling, 45
mask function 面具函数, 112–114
Mastering Algorithms with Perl, 156, 195
Math::BigFloat (multiprecision floating-point library), 205, 290
Math::BigRat module, 216
Math::Complex module, 290
mathematics geeks 数学爱好者, 207
matrix transformations 矩阵变换, 334
max() function in **Scalar::Util** module, 229–230
maxstr() function in **Scalar::Util** module, 229
MAYBE label for HTML elements, 20–21, 217
meaning of parsed input 解析的输入的意义, 250
Memoize module, 45–52, 54, 69
memoization, *see* caching and memoization
memoized clock 带记忆的时钟, 54
merging streams 合并流, 180–182
Michie, Donald, 45
min() function, 229
minstr() function, 229
ML for the Working Programmer, x
monster module 怪物模块, 45
morphology 词法, 239–240
mortgage 按揭贷款, 207–209
Mozilla, 131
MSIE, 131
multiple-list iterator 多列表迭代器, 118
my declaration, 3, 9, 44, 47–48
oddity in semantics of 语法中的古怪, 175
my variable, *see* lexical variables
mystery 神秘, 122

N

\n (newline) character, 241–242
-n command-line option, 98
%N hash for naming parsers, 277–282, 312
nasty regex 令人不快的正则, 191
nasty surprises 令人不快的意外, 93, 152, 223
national security 国家安全, 122
network server 网络服务, 62
newline character 换行符, 174, 242, 294
newsgroups 新闻组, 11, 181
Newton, Isaac, 200
Newton-Raphson method, 200–209
nextval method, 116
NEXTVAL operation, 75, 78–79, 112–114
90-10 rule, 72
nodes 节点, 140, 142–143, 149–152
in AST 抽象语法树中的, 263, 267
in **HTML::TreeBuilder** object, 18–19
of linked list 链表的, 170–175
in local propagation network 本地传播网络中的, 315–317
of notional tree of sentential forms 句式的概念树的, 253–255, 269
in outline structure 大纲中的, 295–296
non-reentrant 不可重入的, 4
nonterminal symbols 非终止符号, 251
normalization of equations 方程的标准化, 341
Norvig, Peter, ix, x
null function 空函数, 16
null pointer 空指针, 109
NULL value in SQL databases, 107, 108
numerals, canonical form for 数字的规定形式, 145

O

O() notation, 67
object constructor methods 对象构造器方法, 48, 53, 122, 123, 335, 345, 346, 352
object identity, testing with == 用 == 测试对象的等同性, 319
object methods, caching in 对象方法里的缓存, 62–65
object-oriented (OO) programming 面向对象式的编程, 17, 48, 62–64
octopuses 章鱼, 53–54, 83, 106
oddity 古怪的东西, 16, 175
odometer method 里程表方法, 86–87, 90
OO (object-oriented) programming, *see* object-oriented programming
open() function, 75
opendir() function, 9
operating system 操作系统, 44
operator overloading, *see* overloading
operator precedence 操作符优先级, 238, 248, 263, 276–277, 291, 311
optimization 优化, 53, 72–73, 112, 134, 138–139, 153, 160, 169, 214
option checking 选项检查, 133

optional item, parser for, 361
orcish maneuver, 70
Orwant, Jon, 156
out of scope 作用域之外, 10, 47–49, 60, 175
outlines 大纲, 294–299
overload module, 238, 312
overloading 重载, 238, 310–314
oxcart 牛车, 72

P

pad 格子图, 48–50
Pair objects, 313
palindromes 回文, 141–142
parabola 抛物线, 200–201
Paradigms of Artificial Intelligence Programming, ix, x
param declaration in Linogram, 373–375
parameter specifications in Linogram, 368
parameters 参数, 373–374
parentheses,
  balanced 平衡的, 191–192
  disambiguating expressions 消除表达式的歧义, 35, 38, 291
  in output of parser 在解析器的输出里, 263, 265
  in regex 在正则里, 243
parse method in **HTML::LinkExtor**, 125
parser function 解析器函数, 368
parser operator 解析器操作符, 260
parser, Perl's, 104
Parser module, 256
parsing 解析, 28, 76, 131, 132, 234, 239–314
  overview 概况, 239
  arithmetic expressions 算术表达式, 261–290
  backtracking parsers 回溯解析器, 304–310
  database-query parsing 数据库查询解析, 299–304
  HTML, 18, 124–125
  in general 一般的, 250–256
  lexers 词法分析器, 239–250
  Linogram specifications, 360–371
  overloaded operators for 重载操作符, 310–314
  recursive-descent parsers 递归下降解析器, 256–261
  regexes 正则, 290–294
partial sums 部分和, 211
partition problem 划分问题, 24–26, 61–62
partitioning 划分, 24–26, 61–62
partitions of an integer 整数的划分, 144–150
  comparing 比较, 148
pathfinder.com, 130
patterns 模式, 85, 88, *see also* rgexes
Paulson, Lawrence, x
penurious college student 穷大学生, 136
performance 性能, 41, 52, 61, 69, 72, 221, 303
period of random-number generator 随机数生成器的周期, 101
Perl 6, 116
Perl interactive debugger; Perl 交互式的除错器, 277
perldoc program, 74
permutations 排列, 2–3, 83–88
persistent caches 持续的缓存, 65–66
pi (π), 212–213
pic program, 375
pipe 管道, 76
plain text 普通文本, 17
plumber 水暖工, 136
plutonium 钚, 62, 75
point, definition in Linogram, 330
polymorphism 多态性, 220
pop, 82
POPDIR directive in configuration file, 30
pos() function, 244
postfix form 后缀式, 36
postprocessing function 后处理函数, 363
PostScript, 330, 331
power series 幂级数, 209–216
  overview 概况, 209–213
  derivatives 导数, 213
  symbolic computation 符号计算, 213–216
power series expansion 幂级数展开, 209–216
power set 幂集合, 156
powers of 2, 176–179
Prague, Defenestration of 布拉格的扔出窗外事件, 134
precedence 优先级, 238, 291, 311
  in arithmetic expressions 算术表达式中的, 263, 277
predicate function 谓词函数, 70
preprocessor 预处理器, 90
primitive features in Linogram, 328
print(), as expression, 283
print statement, 6, 277
printer 打印机, 41
printf() function, 13
priority queue 优先级队列, 149, 151
private variables, *see* lexical variables
probability 概率, 42
productions in grammars 语法中的产品, 251
profiler 剖析器, 72
profiling 剖析, 72–74
Prolog programming language; Prolog 编程语言, 315
promise 断言, 171–172
  forcing 推动, 172, 179
propagation network 传播网络, 324–325
prototypes 原型, 104, 117, 118, 157, 195, 203, 225–229
proxy server 代理服务器, 71
pruning 修剪, 141, 253–255, 269
pseudo-random 伪随机的, 100
pun 双关语, 79, 249
punctuation, excessive 过多的标点符号, 94
pure functions 纯函数, 54
pure virtual methods 纯虚方法, 17
push() function, 82
PUSHDIR directive in configuration file, 30

Q

q{...} syntax, 57
qmail mail system, 196
quantification of regexes 正则的量化, 291
queue 队列, 80–83, 124, 127–128, 133
queue members 队列成员, 127
qw(...) syntax, 98

R

rabbits 兔子, 22–23
rand() function, 100
random() function, 100
random number generation 随机数生成, 100–103
Raphson, Joseph, 200
readdir() function, 77
READLINE method, 123
reciprocal 倒数, 344–345
  of power series 幂级数, 215–216
recursion 递归, 1
  and caching 和缓存, 42–43
  as DFS, 142–144
  elimination of 移除, 152–169, 232
recursive-descent parsers 递归下降解析器, 256–261
  overview 概况, 256
  compound operators 合成的操作符, 259–261
  parser operators 解析器操作符, 258–259

very simple parsers 非常简单的解析器, 257–258
recursive streams 递归流, 175–179, 186, 215
redo, 162
re-entrancy 重入, 4, 31–32
reentrant function 可重入的函数, 32
ref() function, 58, 335–336
ref($base) || $base technique, 335–336
refactoring 重构, 147–148, 156–159, 161, 166–169, 219
reference arguments, functions with 函数的参数是引用, 61
references 引用, 6–7, 13, 27, 40, 46–49, 57–58, 108, 312, 335, 369
  comparing with == 用 == 测试, 111
  from pad 来自格子图, 48–49
  functions returning 函数返回, 53–54
  functions accepting 函数接受, 61
  stringification of 字符串化, 58, 61, 111, 277, 312
referrer, fake, 128
referring URL, 127
regex engine 正则引擎, 77, 315
regex string generation 正则字符串生成, 181–200
regexes 正则, 77, 94, 181, 191, 240, 243–246, 290–294, 315
  as examples of declarative programming 作为声明式编程的实例, 315
  as lexers 作为词法分析器, 243
  syntax of 语法, 291
relaxation 放宽, 326
repetition of parser operation 解析器操作的重复, 260
return-value parameter 返回值形参, 160
reverse() function, 88
RGB, 41
robot policy 机器人政策, 130
robots.txt file, 130–131
Roode, Eric, xi, 117–118
root feature in linogram, 327
root of equation 方程的根, 201
ROOT type in Linogram, 362
round-off error 舍入误差, 204–205, 215
RPN (reverse Polish notation) 逆波兰表示法, 35–36

S
-s operator, 8, 10
/s regex modifier flag, 300
s/// operator, 57
scalar 标量, 6–7, 120–122, 244
  tied 系住的, 121–122
scalar context 标量上下文, 77, 104, 120–121, 126
Scalar::Util module, 226
scaling operation 缩放操作, 181, 214–216, 219–220, 225–226, 336–337, 339, 341, 346–347, 349, 352–353
scanning, *see* lexers
schema 摘要, 91
Schönfinkel, Moses, 218
Schwartz, Randal, ix, xi
Schwern, Michael, 96
scope 作用域, 13, 44, 47–49, 60, 64, 175, 303
  vs. duration 对比有效期, 47–58
searching databases backwards 反向搜索数据库, 97–100
secret weapon, 122
seed (of pseudo-random number generator) 伪随机数生成器的种子, 100
seek() function, 93, 97
self-referent stream definition, 175–179, 186, 215
semipredicate function 半谓词函数, 70
semipredicate problem 半谓词问题, 70, 81, 107, 112
sentences (in grammars) 句子（语法中的）, 251
sentential form 句式, 251
serialization 序列化, 45
set difference 置否, 237
set_prototype() function, 226
shift() function, 82
silly examples 愚蠢的例子, 122
sin() function, 209
sinh() function, 213
Sitaker, Kragen, x
Slashdot, 223
slope 斜率, 201
sort() function, 66, 199, 230
sorting 排序, 54, 66–69
  partitions 划分, 148
  comparator functions 比较器函数, 66–68, 194–195, 233, 236
  tiebreakers in 决胜局, 67
  to obtain canonical form 获得规定形式, 145
soup, fish-dog-and-carrot, 106, 184
spam, 242
special cases 特殊情况, 22, 72, 157, 190, 273, 304, 319, 337, 357, 371
splines 锯齿线, 374
split() function, 89
sprintf() function, 1, 69
SQL, 107, 110, 315
sqrt() function, 200
square, definition in Linogram, 333
square() function, 54
srand() function, 101
srand48() function, 101
srandom() function, 101
stack 栈, 32, 36–37, 298
  call stack 调用栈, 81–82, 152–153
    explicit simulation of 明确的模拟, 25, 161–169
start symbol 开始符号, 251–253
start tag in HTML; HTML 中的开始标签, 17
stat() function, 196
statements as expressions 作为表达式的语句, 283–284
static variables 静态变量, 44
Stein, Lincoln, 88
Storable module, 55, 58
STORE method, 121–122
streams 流, 172, 174, 222, 250
  merging 合并, 180–182
strftime() function, 198
strict 'vars', 230
string 字符串, 63
string generator 字符串生成器, 293
string literals, lexing 字符串字面，词法化, 300–301
stringification 字符串化, 277, 312
Structure and Interpretation of Computer Programs, x
StrVal() function in **overload** module, 312
stub functions 存根函数, 46–47, 49–50, 65, 73
subclass 子类, 17
subdirectories 子目录, 8
sub-parser 子解析器, 250
substr(), 227–228
subtraction as inverse of addition 减法作为加法的逆运算, 322–323
subtype definitions 子类型定义, 362
sum() function, 229–231
Sussman, Gerald Jay, x
Sybase, 96
symbolic links 符号链接, 10, 13, 16, 121
  dangling 悬空的, 13, 16, 121, 217
symbolic references 符号引用, 33, 367
symbol table 符号表, ix, 45–46
symbols in grammars 语法中的符号, 251–253
symbols, terminal vs. nonterminal 符号，终止符 vs. 非终止符, 251

syntactic equivalence of tokens 记号的词法等价物, 291
syntactic sugar 语法糖, 79, 80, 113, 172, 249
syntax 语法, 116, 265
  bizarre 奇异的, 118
  irregularity of Perl's; Perl 的不规则, 303
  obsession with 着迷, 116
synthetic constraints 综合的约束, 347
Synthetic_Constraint_Set class, 348–351
system administrators 系统管理员, 98
systems programming 系统编程, 99

T
$^T special variable, 54
\t (tab) character, 300
tab character, 300
tac command in Unix, 99
_tag key in **HTML::TreeBuilder**, 18
tags 标签, 18, 20–21, 29, 125
tai64n, 196, 197
tail of linked list 链表的尾部, 170
tail-call elimination 尾调用移除, 152
tangent function 正切函数, 213
tangent line 切线, 201
tautology, 339
tell() function, 97
temperature conversion 温度转换, 315–317
terminals 终止符, 251
terminal symbol 终止符号, 251
terminator pattern 终止符模式, 241–243
term of arithmetic expression 算术表达式的项目, 263
terms 项目, 263
$textfunc argument, 20, 38–40
text object in Linogram, 374
thrashing 痛苦的反复, 84
three-dimensional diagrams in Linogram, 371
tie() operator, 65, 114, 121–124
tied hashes 系在/关联散列, 66, 277
TIEHANDLE method, 123
TIESCALAR method, 122
time() function, 54, 73
Time::HiRes module, 73
times() function, 74
to-do list, *see* agenda
tokens 记号, 36–37, 89–90, 239–240
  syntactic equivalence of 词法等价物, 291
tokenization, *see* lexers
token-manufacturing function 记号制造函数, 301
Torkington, Nat, xi, 84
tortoise and hare algorithm 龟兔算法, 206
tourists 游客, 50
Tower of Hanoi 汉诺塔, 4–8
transcendental functions 超越函数, 209
transforms, *see* filters and transforms
treasure 宝物, 24–26, 136–139
trees 树/树型/树状, 8–9, 17–18, 38, 139–140, 143–144, 251–254, 269, 294, 357
  iterator for searching 搜索迭代器, 149–152
  printing 打印, 155, 161–162
  recursion as search of 搜索作为递归, 142–144
  traversal in various orders 以不同的次序遍历, 82
tricks 戏法/技巧/秘籍, 56, 59, 70–71, 78, 89, 104, 208, 248
trigonometry 三角学, 327, 331
trivial equation (0=0) 平凡方程, 336
trivial stream 平凡的流, 172
Tuma, Jan J., 216
tuples in Linogram, 328, 345–347, 351–354, 370–372
Type class, 354
Type::Scalar class, 354–355
%TYPES hash in Linogram, 362
typesetting 排字工人, 135
type specimen catalog 字体样本目录, 135

U
ugly code 丑陋的代码, 248
unary minus 一元的负号, 370
undef, 61, 105–108, 110
undefined variables in calculator program 计算器程序中未定义的变量, 284
union scale 工种级别, 136
UNIVERSAL::isa() function, 58, 119, 172
Unix, 10, 76, 98, 196
epoch time format 纪元时间格式, 197
untagged text 未标记的文本, 18
until loop, 155
upto(), 78–80, 172–173
URI::URL module, 132
URL, 124–128
  bad, 126
  fragment part, 132
  mailto, 132
  referring, 98, 127
user-agent, 131
user parameter to callback function 回调函数的用户形参, 32, 33, 39–40, 247, 258, 303
user interfaces 用户界面, 11
user-supplied key generators, applications of 用户提供的键生成器的应用, 58–59

V
Value class, 343–344
Value::Constant class, 344–345, 372
Value::Feature class, 352–354, 357, 372–373
Value::Tuple class, 346–347
value of parsed input 解析的输入的值, 250
values() function, 77
variables, static 静态变量, 44
VERBOSITY directive, 27
virtual methods 虚方法, 17
vline, definition in Linogram, 332

W
Wadler, Philip, x
Wall, Larry, ix, xi, 96, 116
waste of time 浪费时间, 41, 68
web browsers 网页浏览器, 44, 71
web robots, *see* web spiders
web server log 网页服务器日志, 97–98
web spiders 网页抓取, 124–133
  overview 概况, 124–125
  DFS unsuitable for 深度优先搜索不适合, 141
  pursuing only interesting links 仅追逐有趣的链接, 126–127
  referring URLs 参考 URLs, 127–130
  robots.txt, 130–132
$whence argument to **seek()**, 93
while (caller()) loop, 280
while loop, 9, 123, 138
  reading filehandle in 读取文件句柄, 123
  that executes only once 只执行一次, 147
whitespace 空白符, 18, 294
wildcard 通配符, 88–90
Windows, 76
WIRED magazine 《连线》杂志, 151
"without"operator, 237
wizard 术士, 136
Wolf Book, 156, 195
Wong, Meng, xi
word boundary 单词边界, 181, 293

Wrigley, Ave, 124

`WWW::RobotRules` module, 131–132

`WWW::SimpleRobot` module, 124, 132–133

WYSIWYG drawing system 所见即所得绘图系统, 327

Y

`yes` command in Unix, 76

Yukon Territory, 118

Z

`0e0` trick, 70–71

`"0 but true"`, 71

函数索引

A
a_var(), 338, 341
action(), 296–298
add(), 227, 343
add2(), 210–211, 214–215, 218–219
add_constant(), 342
add_constants(), 344–345
add_constraints(), 359, 368–369
add_declarations(), 362, 363, 365, 368
add_draw_declaration(), 368, 369
add_drawable(), 358
add_equations(), 337, 342, 353
add_feature_con(), 353
add_feature_tuple(), 353
add_features(), 352
add_labeled_constraint(), 349
add_subfeature(), 355, 358, 368
add_tuples(), 347
all_leaf_subfeatures(), 357
allinput(), 247
alternate(), 259–262, 264–268, 271–274, 276, 279, 281–285, 288, 290, 292–293, 302, 303, 305, 308, 309, 311
alternate2(), 311
append(), 106
apply(), 340, 341, 348, 349, 353, 375
apply2(), 349, 350, 353
apply_hash(), 351, 353
arithmetic(), 337–338
AST_to_string(), 38
attach(), 318, 320
attachments(), 318

B
bal(), 192
binary(), 1–2, 160
blocks(), 247

C
c_grep(), 225
c_imap(), 225
c_scale(), 226
callbackquery(), 93–94, 96, 99, 100, 234, 236–238, 299, 302–304
carrots(), 175–176
cgrep(), 222–223
charclass(), 187
check_declarator(), 365
check_move(), 7–8
chronologically, 67–69
cmap(), 222–223
coefficient(), 336
combine(), 229–234
combine2(), 219, 221–222
comparator(), 66–68, 70, 194–195, 233, 236
component(), 346
components(), 346
concat(), 184–190, 192, 291–293
concatenate(), 258, 260–269, 271–276, 279, 281–285, 288, 292–293, 296–298, 302–303, 306–313
concatenate2(), 307, 310–311
constant(), 184, 339
constraint(), 349
Constraint::
  add_constant(), 342, 353
  mul_constant(), 342
  new_constant(), 342
  qualify(), 342
constraint_set(), 343, 356, 359
Constraint_Set::
  constraints(), 343
constraints(), 343, 348, 349, 356
contains_octopuses(), 83
convert_param_specs(), 368–369, 373
cos(), 209, 210, 212
cosh(), 213
curry, 224
Curry::
  curry(), 224–227, 231, 233
  curry_listfunc(), 224
  curry_n(), 224, 227
cut_loops(), 206
cutsort(), 194–195, 200
cwd(), 30, 32

D
dbi_query_iterator(), 110, 112, 114–115
debug(), 279–280
define_config_directive(), 29–31
defines_var(), 341–342
derivative(), 213
DESTROY(), 63
digest_maillog(), 197–198
dir_size(), 15
dir_walk(), 12–17, 19–21, 27, 78, 80–82, 106, 121, 123, 228–229
display_failures(), 288–290
div(), 343
divide(), 216
do_push(), 101
double(), 220, 225
draw(), 359, 362
drawables(), 358, 359
drop(), 174–175, 187, 191, 198, 208–209, 250, 286
duplicate(), 335

E
each(), 120, 233
each_array(), 117–120
eachlike(), 120–121
End_of_Input(), 256–257, 261–264, 266, 267, 270, 281, 283, 287, 295, 306, 308, 309, 313, 362
Environment::subset(), 359
Equation::
  a_var(), 338, 341
  add_equations(), 337, 353
  arithmetic(), 337
  coefficient(), 336
  constant(), 339
  defines_var(), 341
  duplicate(), 335
  is_inconsistent(), 341
  is_tautology(), 339
  new(), 335–336
  normalize(), 341
  scale_equation(), 337, 341
  substitute_for(), 337
  subtract_equations(), 337
  varlist(), 336, 338–339
Equation::System::
  apply(), 340, 341
  equations(), 339–340
  new(), 339
  normalize(), 341
  solve(), 340
  values(), 341
equations(), 339–340
error(), 285, 288, 290
evaluate(), 36–38, 211–212
exp(), 213
Expression::
  new(), 369–370
expression_to_constraints(), 369, 371–373

F
fact(), 216
factorial(), 3–4, 9, 160–161, 210, 213, 216
fastfib(), 50
fastquib(), 50
FETCH(), 121–122, 278
fetchrow_arrayref(), 108, 110, 112–115
fib(), 23, 43–46, 50, 162–165, 167
file_size(), 15
filehandle_iterator(), 79, 91, 114
filter(), 175
find_share(), 24–26, 61–62, 136
FlatDB::
  callbackquery(), 93–94, 302, 304
  new(), 92
  query(), 92–93, 304

FlatDB::Iterator::
  new(), 99–100
  query(), 99
FlatDB::Parser::
  parse_query(), 304
FlatDB_Composable::
  callbackquery(), 236
  query(), 235
  query_and(), 236, 237
  query_not(), 236
  query_or(), 236
  query_without(), 236–237
FlatDB_Ovl::
  callbackquery(), 238
  query(), 238
fold(), 232

G

get_links(), 124–127, 129
goto(), 154, 162
grep(), 21, 80, 82, 103, 104, 106, 174–175, 217, 222, 223

H

hamming(), 179
hanoi(), 6–8, 27
has_component(), 346
has_label(), 349
has_same_components_as(), 346
has_settor(), 319
has_subfeature(), 356
has_value(), 319
head(), 124, 126–127, 129, 172–175, 180, 182, 184, 189–190, 194–195, 206–207, 211, 212, 214–216, 219, 221–222, 250, 257, 286, 287, 289, 297, 298, 306, 308

I

i_and(), 235–236
i_and_(), 234
i_or(), 233–236
i_or_(), 233–234
i_without(), 236, 237
i_without_(), 237
ignore_ignorable_whitespace(), 18
igrep(), 105–106, 131, 175
imap(), 104, 224–225, 230
increment_pattern(), 85–86, 90
index_of_shortest(), 189–190
insert(), 194–195
interesting_files(), 82–83, 106
intrinsic(), 352
Intrinsic_Constraint_Set::
  apply(), 348
  constraints(), 348
  new(), 348
  qualify(), 348
  union(), 348
intrinsic_constraints(), 357
Investor::total(), 63–65
is_hamming(), 179–180
is_inconsistent(), 341
is_promise(), 172
is_scalar(), 355
is_special(), 110–112
is_tautology(), 339
iterate_function(), 87, 171, 175, 197, 203, 204, 211, 221
Iterator(), 116
Iterator2Handle::
  READLINE(), 123
  TIEHANDLE(), 123
Iterator2Scalar::
  FETCH(), 122
  STORE(), 122
  TIESCALAR(), 122
Iterator_Logic::
  i_and(), 235–236
  i_and_(), 234
  i_or(), 233–236
  i_or_(), 233–234
  i_without(), 236, 237
  i_without_(), 237
iterator_to_stream(), 250
Iterator_Utils::
  append(), 106
  filehandle_iterator(), 91
  igrep(), 105–106
  imap(), 104
  iterate_function(), 87
  Iterator(), 78–80
  list_iterator(), 106
  NEXTVAL(), 79

K

kindof(), 343–346, 352, 353, 359

L

labels(), 349
length_and_product(), 232
level_of(), 297–298
Lexer::
  allinput(), 247
  blocks(), 247
  iterator_to_stream(), 250
  make_charstream(), 240
  make_lexer(), 249
  records(), 240–242
  tokens(), 243–245
List::Util::reduce(), 230, 231
list_iterator(), 106
list_of(), 260
list_to_stream(), 171, 194–195
literal(), 182, 185, 187–190, 291–293
log(), 213
lookfor(), 257–262, 264–268, 271–272, 274–277, 279, 281, 283–288, 290, 292–293, 296, 297, 301–303, 306, 308, 309, 311, 314, 325, 361, 364, 370

M

make(), 318
make_charstream(), 240
make_dfs_search(), 149–152, 254, 255
make_lexer(), 249, 262, 283, 291, 295, 300, 360
make_rand(), 102–105
maketoken(), 246–248
map(), 174, 222–224, 230
matches(), 191
max(), 229–231
maxstr(), 229
memoize(), 46–47, 57, 60, 65, 66, 68, 151, 217
merge(), 180–181, 234
min(), 229, 231
mingle2(), 182
minstr(), 229
mul(), 343
mul2(), 211, 213, 214, 219
mul_constant(), 343
mul_constants(), 344, 345
mul_feature_con(), 353
mul_tuple_con(), 347
multiply(), 214–216

N

name(), 53, 319, 320
negate(), 343, 344
new(), 48, 54, 63, 93, 94, 98–100, 125, 131, 205, 216, 234, 290, 320–324, 335–337, 339, 342, 345–350, 352–354, 356–360, 362, 363, 366, 369–370, 372
new_adder(), 321–325
new_constant(), 323–325, 342
new_from_var(), 352, 372, 373
new_multiplier(), 322–325
NEXTVAL(), 78–79, 112–117, 129–130, 224
no_such_directive(), 34
node(), 171–176, 180, 182–184, 187, 189–190, 192, 194, 198, 203, 206–207, 211, 214–216, 219, 221–222, 250, 308
Node::
  name(), 320
  new(), 320, 321

notify(), 320
revoke_wire(), 320
set_wire(), 320
wire(), 320
normalize(), 341
nothing(), 273, 288, 306, 309
notify(), 320
notify_all_but(), 318–319
null_list(), 273, 282

O

Octopus::
name(), 53
new(), 53–54
op(), 343–344
open_input_file(), 33–34
operator(), 275–277, 284, 301, 310, 325, 369, 370
owed(), 208–209

P

parent, 355
parse_query(), 304
parse_section(), 76
parse_section_title(), 76–77
parse_variables(), 76–77
parser(), 258, 311
Parser::
action(), 296–298
alternate(), 259–262, 264–268, 271–274, 276, 279, 281–285, 288, 290, 292–293, 302, 303, 305, 308, 309, 311
concatenate(), 258, 260–269, 271–276, 279, 281–285, 288, 292–293, 296–298, 302–303, 306–313
End_of_Input(), 256–257, 261–264, 266, 267, 270, 281, 283, 287, 295, 306, 308, 309, 313, 362
error(), 285, 288, 290
list_of(), 260
lookfor(), 257–262, 264–268, 271–272, 274–277, 279, 281, 283–288, 290, 292–293, 296, 297, 301–303, 306, 308, 309, 311, 314, 325, 361, 364, 370
nothing(), 273, 288, 306, 309
operator(), 275–277, 284, 301, 310, 325, 369, 370
star(), 186, 190, 276–280, 282–283, 291, 293, 295–296, 298, 301, 313, 361–362, 370
T(), 271–273, 281, 296–298, 303, 307, 311, 313
test(), 297
Parser::Debug::
alternate(), 281–282
concatenate(), 279
debug(), 280
lookfor(), 281
star(), 282
T(), 281
partial_sums(), 211
partition(), 25, 136, 137, 141, 144–145, 147
pattern_to_permutation(), 85, 87
permute(), 84–85, 87
plus(), 187, 188, 190, 291, 293
postcat(), 184, 189
pow2_from(), 176, 221
powers_of(), 211
PowSeries::
add2(), 211
derivative(), 213
divide(), 216
evaluate(), 211
mul2(), 211
multiply(), 215–216
partial_sums(), 211
powers_of(), 211
recip(), 215
sum(), 215
term_values(), 211
precat(), 184, 189
print_tree(), 155, 161–162
promise(), 172
promote_if(), 22, 218
promote_if_h1tag(), 21, 217–218

Q

qualified_intrinsic_constraints(), 357
qualified_synthetic_constraints(), 358
qualify(), 342, 348, 349, 356–358
query(), 92, 93, 99, 234–236, 238, 293, 299, 302, 303
query_and(), 235–238, 299, 301
query_not(), 235–237
query_or(), 235, 236, 238, 299, 301
query_without(), 235–238
quib(), 50

R

Rand(), 100–102
read_config(), 27–28, 30–35
READLINE(), 123
recip(), 215
reciprocal(), 343–345
records(), 240–242
reduce(), 229–234
Regex::
charclass(), 187
concat(), 184–186, 293
literal(), 182, 292–293
matches(), 191
plus(), 187, 293
query(), 293
show(), 187
star(), 186, 190, 292
union(), 183, 189
revoke(), 319, 321, 324
revoke_wire(), 320
RGB_to_CMYK(), 41–42

S

scale(), 181, 211, 214, 216, 219–220, 344–347, 349, 352–353
scale_equation(), 337, 341
score_match(), 35
seek(), 93
set(), 318–320, 323, 324
set_wire(), 320, 321
settor(), 319
settor_is(), 318–319
show(), 174, 176–178, 181, 182, 184, 185, 187, 190, 198, 204, 206, 212, 213, 215, 216
simple_query(), 301–303
sin(), 209–210, 213
single(), 308–309
sinh(), 213
slope(), 204, 213, 220–221, 223
solve(), 204–205, 208–209, 212–213
sort comparator(), 67
sort(), 54, 194, 199, 230
special(), 110–112
special_imap(), 111
special_to_undef(), 112
splice(), 84, 85, 151, 195
sqrt(), 37, 200, 205, 209, 331
sqrtn(), 203
SRand(), 101
star(), 186, 190, 276–280, 282–283, 291, 293, 295–296, 298, 301, 313, 361–362, 370
stat(), 196
STORE(), 121–122, 277
Stream::
cut_loops(), 206
cutsort(), 194–195
drop(), 174
filter(), 175
head(), 172
is_promise(), 172
iterate_function(), 175
list_to_stream(), 194
merge(), 180
node(), 172
promise(), 172
show(), 173–174
tail(), 172, 177

transform(), 174–175
upfrom(), 173
upto(), 172–173
sub(), 343, 373
subfeature(), 355–356, 359, 372–373, 375
subfeatures(), 358
subset(), 360
substitute_for(), 337, 340
subtract_equations(), 337
sum(), 215, 230
sunion(), 308, 310
synthetic(), 352
Synthetic_Constraint_Set::
add_labeled_constraint(), 349
apply(), 349
apply2(), 349
apply_hash(), 351
constraint(), 349
constraints(), 349
has_label(), 349
labels(), 349
new(), 348
qualify(), 349, 358
scale(), 349
synthetic_constraints(), 357

T

T(), 271–273, 281, 296–298, 303, 307, 311, 313
tai64n_to_unix_time(), 197
tail(), 172, 177, 180, 182–184, 189, 197, 206–207, 211–216, 219, 221–222, 257, 287, 289, 306, 308
term_values(), 210, 211
test(), 297
TIEHANDLE(), 123
TIESCALAR(), 122
to_hash(), 346
tokens(), 245–249
total(), 63–65
total_size(), 8–11, 14, 17, 27
transform(), 174, 177, 180, 197, 198, 203, 210, 221, 225
traverse(), 126, 128, 129, 132
trim(), 296, 298
Type::
add_constraints(), 359
add_drawable(), 358
add_subfeature(), 358
all_leaf_subfeatures(), 357
constraint_set(), 356
constraints(), 356
draw(), 359
drawables(), 358
has_subfeature(), 356
intrinsic_constraints(), 357
is_scalar(), 355
new(), 354
parent, 355
qualified_intrinsic_constraints(), 357
qualified_synthetic_constraints(), 358
subfeature(), 355–356, 372
subfeatures(), 358
synthetic_constraints(), 357
Type::Scalar::
add_constraint(), 355
add_subfeature(), 355
is_scalar(), 355

U

undef_to_exhausted(), 113
undef_to_special(), 111–112
union(), 183–185, 187–190, 192, 291–293, 308, 309, 348, 353
untag_html(), 19–20
upfrom(), 173–176, 210, 213, 221
upto(), 78–80, 106, 120, 172–174
upto_list(), 172–173

V

value(), 319, 320, 345
Value::
add(), 343
div(), 343
mul(), 343
negate(), 343, 344
op(), 343–344
reciprocal(), 343, 344
sub(), 343
Value::Constant::
add_constants(), 345
kindof(), 345
mul_constants(), 345
new(), 345, 372
reciprocal(), 344, 345
scale(), 345
value(), 345
Value::Feature::
add_feature_con(), 353
add_feature_tuple(), 353
add_features(), 352
intrinsic(), 352
kindof(), 352
mul_feature_con(), 353
new(), 352
new_from_var(), 352, 372, 373
scale(), 352
synthetic(), 352
Value::Tuple::
add_tuples(), 347
component(), 346
components(), 346
has_component(), 346
has_same_components_as(), 346
kindof(), 346
mul_tuple_con(), 347
new(), 346, 372
scale(), 346
to_hash(), 346
values(), 341
varlist(), 336, 338–339

W

walk_html(), 19–22, 38–40, 217–218
wire(), 320, 324
Wire::
attach(), 318, 320
attachments(), 318
has_settor(), 319
has_value(), 319
make(), 318
name(), 319
new(), 317
notify_all_but(), 318–319
revoke(), 319, 321
set(), 318
settor(), 319
settor_is(), 319
value(), 319