Index

Symbols

--> 47
:: 107
::= 3
$E[v \to E_1]$ 145
$[\![\ ]\!]$ *See* emphatic brackets
$\perp$ 346
$\to$ 356
ε 18
λ 140
| 18
• 18
$\subseteq$ 345
$\Leftrightarrow$ 149
$\Leftrightarrow^*$ 149
$\Rightarrow$ 5, 148
$\Rightarrow^*$ 148
$\Rightarrow_\alpha$ 148
$\Rightarrow_\beta$ 148
$\Rightarrow_\delta$ 150
$\Rightarrow_\eta$ 149
➨ 262
➞ 246
➞ ∞ 260

A

abstract data types 443, 471
abstract productions 24–26
abstract syntax 23–29, 240–242, 273, 481–485
  lambda calculus 162
abstract syntax trees 22, 92, 161, 275, 279
  Prolog structure 26–27
action semantics 507–564
  abstractions 546–547
  actions 508
  basic facet 515
  Bindable 534, 541, 543
  calculator 522–530
  cells 518
  current information 514
  Data 513
  data 508
  data and sorts 511–514, 534, 543
  Datum 513
  declarative facet 515, 531–534
  empty bindings 531
  functional facet 515
  hybrid actions 515
  imperative facet 515, 518
  join 513
  lower level 511
  meet 513
  merge bindings 532
  modules
    Declarative 531
    Imperative 522
    Integers 512
    TruthValues 512
  nothing 513
  overlay bindings 532
  Pelican 541–559
    semantic equations 552–554
  Procedure 546
  procedures 547–549
  reflective facet 515, 545
  stable data 519
  Storable 518, 534, 543
  subsort 513
  translating to actions 551–557, 559–563
  upper level 511
  Value 534, 543
  Wren
    semantic equations 535–536, 537–540
    semantic functions 534–535

yielders 508
abstraction of _ 546
application of _ to _ 548
closure of _ 546
the _ bound to _ 510, 515, 532
the _ stored in _ 509, 515, 519
the given _ 510, 514
the given _ # _ 514
actions 508, 515
combinators
_ and _ 516, 532
_ and then _ 516, 533
_ before _ 542
_ hence _ 533
_ moreover _ 541
_ or _ 510, 536
_ then _ 510, 517, 533
_ thence _ 548–549
composite actions
unfold 538
unfolding _ 537–538
hybrid actions
recursively bind _ to _ 550
primitive actions
allocate a cell 509, 519
bind _ to _ 509, 531
check _ 537
complete 509
enact _ 547
fail 509
give _ 509
rebind 541
regive 526
store _ in _ 509, 519
admissible predicates 383
ADT *See* abstract data types
affix 253, 295, 354
algebraic semantics 443–506
abstract syntax 481–485
canonical form 479
carrier sets 462
completeness 468–470
conditional equations 448, 451–452
confusion 451, 467, 485
congruence 463–464
consistency 469
constructors 450–451, 481
equations 444
error values 450
evaluation function 463
function syntax 448
ground terms 461
implementation 499–506
implementing queues 474–486
initial algebras 451, 467
junk 451, 467
mathematical foundations 460–470
models 464
module syntax 447–448
modules 445–459
AbstractExpressions 484
Arrays 474
Booleans 446, 470, 500–501
Bools 460
Characters 452–453
Expressions 482
Files 455
Lists 453–454
Mappings 456–458, 459
Mixtures 459
NatLists 486–487
Nats 460
Naturals 449–450, 501–502
Predecessor_1 468
Predecessor_2 468
Queues 472–473
Stores 458
Strings 456
normal form 479
parameterized modules 453–456
quotient algebras 465–466
representation function 475
Σ-algebras 461–463, 477–481
Σ-homomorphisms 466
selectors 481
signatures 444
sorts 444
sufficiently complete 481, 488

term algebras 462
type of interest 478
Wren 487–499
Wren interpreter 494–498
Wren modules
WrenASTs 489–490
WrenEvaluator 495–498
WrenSystem 498
WrenTypeChecker 490–494
WrenTypes 488
WrenValues 489
Allison, Lloyd 339
ambiguous grammars 6, 12
expressions 16, 18
$a^nb^nc^n$ 8, 19, 60–66, 138
anonymous function 140
antisymmetric 345
ascending chains 346
Astesiano, Egidio 242, 270
attribute grammars 59–104
binary numerals 67–71
binary trees 73
code generation 191–222
definition 66–67
expression language 73
inherited attributes 63
semantic rules 66
semantics 67–73
synthesized attributes 61
Wren
auxiliary functions 89
commands 80–82
context checking 74–91
declarations 75–80
expressions 82–84
procedures 91
semantic rules 84–89
translation semantics 191–215
auxiliary functions 313–315
axiomatic semantics 395–442
assertion 396
factorial 405–407
limitations 396
loop invariants 405, 408–410
nested while loops 410–414
partial correctness 397
Pelican 418–432
axiomatic definition 428–429
blocks 420–421
constant declarations 420
nonrecursive procedures 422–424
procedure declarations 420–421
recursive procedures 425–429
postcondition 397
precondition 397
prime numbers 414
program derivation 438–442
proof tree 404
restrictions 423–424, 431–432
table of values 409, 414
termination 432–436
indefinite iteration 432
recursion 435
total correctness 397
Wren 398–418
assignment 398–400
axiomatic definition 408
input/output 400–401
while command 405–407

B

Backus, John 30
Backus-Naur Form *See* BNF
big-step semantics 261
binary search 441
BNF 2, 3
bottom 345

C

'C' 45
calculator language 310
abstract syntax 279, 523
action semantics 522–530
concrete syntax 277–278
example 281, 528–530
semantic equations 282,

524–528
semantic functions 282, 523–524
semantics 280
call by name 155
call by value 155
canonical form 479
Cantor, Georg 343
cardinality 343
carrier sets 451, 462
Chomsky, Noam 2, 30, 59
Church, Alonzo 139, 153–155
closure 229
Colmerauer, Alain 57, 566
command continuations 334
compilers 187, 224–225
complete partial order 346
completeness 250–252, 468–469
composition 282
compositional 272, 276, 293
concrete operational semantics 224
concrete syntax 21, 50, 481–482
conditional **and** 252, 386, 541, 603
conditional equations 448, 451–452
conditional expressions 237, 390
confusion 451, 467
congruence 463–464
connect (Prolog predicate) 45
consistency 251–252, 469
constants, declarations of 318, 509–510, 543
constructors 450–451, 481
context checking
attribute grammars 74–91
denotational semantics 323–327
two-level grammars 116–132
context conditions 13, 15, 324
context-free
grammars 3, 4, 388
syntax 14
context-sensitive
errors 78–80, 95
grammars 3, 8, 60–66
syntax 14
continuation semantics 328–331
Gull
semantic domains 335
semantic equations 336–338
semantic functions 335
continuations 331–338
Continuity Theorem 377, 387
continuous functionals 374–378
composition 376
conditional expression 375
constant function 375
identity function 375
continuous functions 356, 361–362
cpo *See* complete partial order
current information 514
curried functions 143–144, 283, 288, 608–609
Curry, Haskell 144
cut 56

D

dangling else 12
data 508
data abstraction 471–472
Datum 513
deallocate storage 321
declaration continuations 334
declarative languages 565
definite clause grammars 57
See also logic grammars
denotable values 311, 313
denotation 271
denotational semantics 271–340
auxiliary functions 274
calculator language 277–285
context checking
semantic equations 326–327
errors 329
numerals 274
Pelican 311–323
abstract syntax 311

environments 313
procedures 318–321
semantic domains 315
semantic equations 316–318
semantic functions 316
stores 314–315
propositional logic 304
semantic domains 273
semantic equations 274
semantic functions 273–274
Wren 285–304
auxiliary functions 290
context checking 323–328
errors 293–294
example 296–302
input and output 294–304
prototype interpreter 304–310
semantic domains 286–287
semantic equations 290–293
semantic functions 289
dependency graph 7
derivation trees 5, 19–20, 44, 113–115, 483
diagonalization 343
difference lists 44–45, 57
Dijkstra, Edsger 442
direct denotational semantics 293, 328
direct implementation 480
disjoint union 287, 311
divides relation 347
domain theory 345–365
A+B 351
AxB 349
A→B 356
approximates 346
ascending chain 346, 356
bottom 345
complete partial order 346
continuous functions 356, 361–362
elementary domains 348–349
Fun(A,B) 355
function domains 355–361
improper value 348
injection 352
inspection 352
less defined than or equal to 346
monotonic 355
product domain 349–351
projection 350, 352
sum domain 351–354
dynamic errors 16
dynamic scoping 177–178, 234, 319, 603

E

E* 18
E^+ 19
$E^?$ 19
elementary domains 348–349
emphatic brackets 272, 528
E^n 19
encapsulation 472
English grammar 4–8, 40–41
generating sentences 46
parsing 43–45, 47–48
environments 170–174, 228–231, 235–236, 288, 311–312, 314, 324–325, 388
error continuation 336–338
evaluation function 463
expressible values 288, 313
expression continuations 334
expression language
abstract syntax 483–485
concrete syntax 482
extensionality 150

F

facets 514, 515
finite functions 246, 456–458, 522
first-class values 288, 606
fix 373, 392–393
fixed point 368–369
Fixed-Point Identity 373, 393
fixed-point induction 383–384
fixed-point semantics 365–388

admissible predicate 383
approximates 366
approximations 373
constructing a fixed point 370–373, 378
continuous functionals 374–388
fixed-point induction 383–384
less defined than or equal to 366
natural extension 366, 367
nonrecursive functions 379
while command 380–382
Fixed-Point Theorem 370
flat domains
See elementary domains
Floyd, Robert 395, 442
formal specifications 507
free word algebra 478
front end 31–33
funarg problem 177–178
function domains 355–361
function subprograms 323
functional facet 515–518
functional programming 139, 158, 588
functionals 368
functions 355

G

generic types 472
Gordon, Michael 339
goto commands 329, 396
grammars 2
Gries, David 438
ground terms 461
Gull 333–338
abstract syntax 333
semantic equations 336–337
semantic functions 335
Gull programs
fact 338
labels 334
Guttag, John 480, 506

H

halting problem 155, 355
Hasse diagrams 345, 363–364
heterogeneous algebras 462
Hoare, C. A. R. 395, 442
Hollerith literals 71, 111–115, 133–136
homogeneous algebras 462
homomorphisms 276, 466, 475, 485

I

imperative languages 285–286, 565, 587
in parameter 321
in-out parameter 321
induction 435
inessential ambiguity 19, 241
inference rules 238, 401
infinite sum domain 353
information hiding 471, 472
inherited attributes 67
initial algebras 451, 467
injection 352
input/output 253
inspection 352
interpreters 225
isomorphism 466

J

join 513
junk 451, 467

K

Kahn, Gilles 261, 269
keywords 10
Kleene closure 18
Knuth, Donald 59, 103
Kowalski, Robert 57

L

λ 140
label expressions 237
labels 331
laboratory activities 31–58
  algebraic semantics 499–506
  attribute grammars 215–222
  context checking 92–103
  denotational semantics 304–310
  Hollerith literals 133–136
  lambda calculus evaluator 160–166
  recursion in lambda calculus 389–394
  SECD machine 235–238
  self-definition
    Prolog 179–185
    Scheme 169–179
  structural operational semantics 264–269
  translating to actions 559–563
  translational semantics 215–222
  two-level grammars 132–133
lambda calculus 139, 341
  abstraction 141
  applicative order reduction 152
  applied lambda calculus 141
  β-abstraction 149
  β-conversion 149
  β-redex 148
  bound variables 145
  Church-Rosser Theorems 153–155
  closed expression 145
  combination 141
  confluence property 154
  equal expressions 149
  equivalent expressions 149
  extensionality theorem 150
  free variables 145, 163
  innermost expressions 152
  lambda notation 140
  lists 156–157
  name clash 145
  normal form 151
  normal order reduction 152, 164
  notational conventions 142
  numerals 157
  outermost expressions 152
  paradoxical combinator 391–392
  parser 161–162
  parsing expressions 143
  pure lambda calculus 156–157
  recursion 389–394
  reduction 145
  reduction rules 147–150
    α-reduction 148
    β-reduction 148
    δ-reduction 150
    η-reduction 149
  reduction strategies 151–155
  scanner 160–161
  semantics 145–146, 151–155
  substitution 146, 163
    safe substitution 145
    valid substitution 145
  syntax 140–141, 160
  variable capture 145
  variables 141
  Y 391
lambda expressions
  Add 157
  Cube 388
  Curry 144, 159
  Double 166
  Fac 390, 392
  Factorial 392
  FourthPower 144
  Head 156
  in Scheme 599
  Nil 157
  numerals 157
  Pair 156
  Sqr 388
  Succ 157
  Tail 156
  Thrice 166, 238, 388
  Twice 142, 160, 166, 238
  Uncurry 144, 159

Y 391
Landin, Peter 223, 270
lattices 347
least upper bound 345
left factoring 55–56
left recursion 52–54, 101, 575
let expressions 158, 172–173, 233, 237, 604
lexical analyzer 12, 21
lexical syntax 10
lexicon 12
Limit Lemma 357
Lisp *See* Scheme
history 588
lists 17, 27, 156–157, 165, 244–245, 295, 343, 354, 444–445, 453–454, 575–579, 591–592
logic grammars 40–50
counting words 49
English 47, 48
lambda calculus 162
motivating 41–44
parameters 47–48
preprocessor 46–47
Prolog code 49
Wren 50–57
logic programming 565–586
See also Prolog
and two-level grammars 136–137
logical variable 400, 582–583
loop 382
loop invariants 405, 433
lower bound 345
lub *See* least upper bound
Lub Lemma 357

M

macrosemantics 511
many-sorted algebras 462
mathematical semantics 271
McCarthy, John 30, 167, 186, 225, 588
meet 513
merging bindings 532
metacircular interpreters 167, 169–174, 225
metalanguage 2, 3, 341
microsemantics 511
ML 140, 244, 564
models 464, 484–485
modules 445–459
monotonic 355
Mosses, Peter 394, 507, 563–564
multiple declarations 97, 120

N

natural deduction 238
natural extension 366, 367
natural semantics 261–262
Naur, Peter 30
nonprocedural languages 565
nonrecursive functions 379
nonterminal symbols 2
nontermination 255–256, 344
normal form 479
normal order reduction 234
nothing 513

O

one-level grammar 105
operational semantics 223–270
overlaying bindings 325, 532

P

Pagan, Frank 138, 185, 222, 339
paradoxes 365
paradoxical combinator 391–392, 394
parameter passing 232
parameterized modules 453–456, 472
parametric polymorphism *See* polymorphic operations
parse trees 5
parsers 21, 31, 92, 161–162

partial correctness 397
partial functions 366
partial order 345
pass by value 232
Pelican
  abstract syntax 311, 419
  action semantics 541–559
  axiomatic semantics 418–432
  axioms and rules 428–429
  context checking 326–327
  context constraints 324
  denotable values 541, 543
  denotational semantics 311–312
  expressible values 543
  procedures 545–551
  storable values 543
Pelican programs
  action 555
  facproc 559
  facwhile 559
  recurrence 431
  scope 316–318
  small 560
  squaring 419
  sum 430
  summation 320–321
  trace1 322
  trace2 322
period 567
phrase-structure syntax 10
Plotkin, Gordon 238, 269
polymorphic operations 141, 144
postcondition 397
power set 343
pragmatics 1–2
precondition 397
procedures 318–321
product domain 349–351
productions 2
program derivation 437–442
projection 352
Prolog 565–586
  anonymous variables 577
  arithmetic expressions 584
  atoms 566
  BNF syntax 568
  body of a clause 567
  clause order 574
  clauses 567
  closed world assumption 570
  comment 33
  constants 566
  control 574–575
  cut 56
  example 569–571
  fact 567
  failure 181
  family tree 573
  function symbols 566
  functors 566
  goal 567
  goal order 574, 582
  head goal 567
  input/output 585–586
  instantiation 582
  lists 575–580
  logical variable 582–583
  not 571
  numbers 566
  numerical comparisons 583
  period 567
  predicates 566
  procedures 572
  query 567
  recursion 572–574
  rule order 574
  rules 568, 570
  scope 570
  search trees 579–580
  sorting 581–582
  structured objects 567
  structures 566
  term construction 585
  unification 571, 583
  unit clause 567
  variables 566
Prolog predicates
  abort 586
  buildSymbolTable 99
  'C' 45

call 586
clause 180
compute 164, 237, 246, 268, 307–308, 390
concat 578
delete 577
get0 34, 585
getfilename 39
go 39, 56, 269, 311, 390
is 584
lookupType 97
member 577
name 586
nl 586
pretty-print 165
prove 180–181
read 586
tab 586
tableIntersection 97
union 578
write 586
prototyping 31

Q

quotient algebra 465–466

R

rand 141
rator 141
recursion 319, 341–343, 368, 550–551
referential transparency 588
reflexive 345
regular expressions 18
regular grammars 4, 33–34
relational languages 565
remainder of the program 331
representation function 475
reserved words 10–11, 15
Robinson, Alan 566
Roman numerals 50, 72, 277
Ross, Peter 186
Roussel, Philippe 566
rule schemes 248
rules of inference 238–239, 401, 420

S

Σ-algebras 461–463
scanners 12, 21, 31, 33– 40, 92, 160–162
Prolog predicates 35–39
Scheme 140, 167–169, 587–610
atoms 589
Cambridge Polish Form 592
composition 607
conditional expressions 597
correctness 605
curried functions 608–609
defining functions 596
dotted pairs 589
empty list 591
evaluation 593
higher-order functions 606–608
lambda expressions 169, 599
let expressions 168, 604
list processing 599–603
lists 591–592
metacircular interpreter 169–174
micro-apply 174
micro-eval 172
micro-evalcond 172
micro-evallet 173
micro-let-bind 173
micro-rep 170
numeric functions 592, 593
predicates 593
recursive definitons 598
S-expressions 589–590
scope rules 603–605
special forms 596
syntax 590–591
tail recursion 609–610
Scheme functions
atom? 168
car 168, 590, 592
cdr 168, 590, 592

concat 168, 601
cond 168, 597
cons 168, 591, 592
define 594, 596
display 168
eq? 595
equal? 168, 601
filter 608
if 598
map 607
member 602
newline 168
null? 168, 595
pair? 595
quote 168, 594
reduce 607
Schmidt, David 339, 394
Schönfinkel, Moses 144
scope 288, 311
scoped data 514
Scott, Dana 271, 344, 394
SECD machine 223, 228–234
configuration 228
control 228
dump 228, 229, 236
environment 228
stack 228
transition function 229–230, 236
selectors 481
self-definition 167–186, 225
function application 174–175
Lisp 167–169
Prolog 179–185
proof tree 180
tracing 181
semantic equivalence 260–261, 294, 395, 551
semantic errors 16
semantic prototyping 339
semantics 1–2, 273–274
sequencers 330
side effects 525, 588
signatures 444, 478
small-step semantics 261
specifications of algorithms 397–398
stable data 514
standard semantics 331
start symbol 3
static scoping 177–178, 229, 233–234, 319, 603
static semantics 14, 30
stationary chains 365
Stepney, Susan 339
Storable 518
storable values 287, 313
stores 245–246, 265–266, 306–307, 314–315, 330, 458, 520
Stoy, Joseph 339
Strachey, Christopher 271
strict functions 517, 367
structural induction 242–245, 276, 476, 479
structural operational semantics 238–270
abstract syntax 239–242
axiom 238
commands 253–264
examples 248–250, 256–260
expressions 245–253
inference system 247
outcomes 250, 255
transition function 247, 254, 265–267
stuck 246
subsorts 513
Substitution Property 464
sufficiently complete 481
sum domain 351–354
symbol table 74–80
symbolic tokens 36–37
syntactic analyzer 21
syntactic categories *See* syntactic domains
syntactic domains 26, 239–240, 272–273
syntax 1–2, 272–273
syntax trees *See* abstract syntax trees

synthesized attributes 66–67
Synthesizer-Generator 222

T

table of values 440
tags 351, 352
tail recursion 384, 609–610
Tennent, R. D. 339
term algebra 462
terminal symbols 2
termination 432–436
termination expression 433
tokens 10–12
  recognizing 37–38
total correctness 397
transient data 514
transitive 248, 255, 345
translational semantics 187–222, 224–226, 551–557
  attribute grammar 207–213
  attributes 192
  commands 201–207
  comparisons 199–201
  expressions 193–201
  labels 202–207, 218
  machine language 188–189
  optimization 196–199
  program translation 189–191
  Prolog implementation 215–222
  temporary variables 190, 193, 199, 220
T_Σ 462
Turing, Alan 155
Turner, S. J. 136
two-level grammars 105–116
  and logic programming 136–137
  derivation trees 113–115
  EMPTY 110
  Hollerith literals 111–112, 133–136
  hyper-rule 106, 110
  lists 110–111
  metanotion 106, 110
  metarule 106, 110
  notational conventions 106–107
  protonotion 105, 110
  representation tables 109, 112
  TALLY 110–111
  terminal symbols 105
  where clause 114
  Wren comands and expressions 124–126
  Wren declarations 117–124
  Wren grammar 106–108
  Wren specification 129–131
type of interest 478

U

unbound 311
unbounded arrays 474–476
unbounded queues 472–473
undefined 289, 312
unparse 29
unrestricted grammars 3
upper bound 345

V

van Wijngaarden, Aad 105, 138
VDL
  See Vienna Definition Language
vending machine 303, 541
Vienna Definition Language 226–227

W

Warren, David 58
Watt, David 103, 339, 506, 507, 564
well-founded partial order 432
while command 380–388, 405, 537–539
white space 36
Wren 10–16
  abstract syntax 26, 28, 239–242, 286, 489–490
  action semantics 531–541

  algebraic semantics 487–499
  attribute grammar 74–91
  axiomatic semantics 398–418
  axioms and rules 408
  BNF 11, 106
  code generation 191–215
  context checking 84–89,
    116–132, 323–328, 490–494
  context constraints 14, 74, 494
  denotable values 534
  denotational semantics 285–304
  expressible values 534
  interpreter 494–498
  logic grammar 50–57
  parsing 50–57
  reserved words 10–11, 37
  semantic errors 16
  storable values 534
  tokens 38
  translational semantics 207–213
  two-level grammar 116–132
Wren programs
  bool 303
  frombinary 500
  gcd 189, 215
  illegal 13
  mod 213
  multiply 191, 214
  prime 305
  prog1 93
  prog2 94
  switch 32
  tobinary 264

Y

Y 391
yielders 508, 514–515