Author and Concept Index

- κ, 258
- σ, 291, 379
- τ, 237, 261, 291, 379
- τ_f, 237
- _*-name_, 113, 257, 285
- _C_, 305
- _*-name_, 9
- _def_, 113, 257, 285
- _exp_, 113, 257, 285
- _formals_, 113, 257, 285
- _integer_, 9
- _type-formals_, 257

- _A_, 721
- $\mathcal{A}$
  - as automatically generated function in identifier cross-reference, 23
- Abelson, H., 151, 164, 379
- Abelson, Harold, 63
- abstraction, functional, 21
- accumulating parameters
  - method of, 71
- _ad hoc_ polymorphism, 101
- Ada, 6, 346, 378, 379
- _add_
  - real Clu function, 374
- `addtoend`, Prolog predicate, 545
- Aho, A., 722
- Algol-60, 64
  - own variables, 92
- Algol 60, 6
- Alphard, 346, 378, 379
- answer
  - nonstandard English in Smalltalk, 403
- Appel, A., 206
- `append`
  - Prolog predicate, 546
- arithmetic
  - mixed, 425
- arity
  - in μSmalltalk, 408
- _array-element_, 408
- association list, §74, 402, 421, 547
  - in sets, 102
- Atkinson, R., 377
- atom
  - μScheme concept, 69

- $\mathcal{B}$
  - as basis function in identifier cross-reference, 23
- Baker, H.G., 205
- Barnes, J.G.P., 379
- Barrett, W., 722
- _binary-functor_, 528
- _binary-predicate_, 528
- Birtwistle, G., 379, 506
- Bishop, J., 379
- blocks world, §559
- Bobrow, D., 505
- bound variable, 225
- Budd, T., 502, 505, 506
- Budd, Tim, 416
- BugInTypeInference[[BugInTypeInference]]
  - in μML, 294
- Burge, W.H., 151
- Burstall, R., 151

- C
  - scope rules, 9
- C
  - own variables `static`, 92
  - scope rules, 55
- C (programming language), 6
- C++
  - overloading, 504
- `c{a,d}`$^{+}$`r`, Scheme functions, 67
- calculational proof, 77
- `call/cc`, real Scheme function, §150
- _car_, real Clu function, 377
- careless persons
  - misusing unspecified values, 605

For COMP 105, Tufts University, Spring 2014 only --- do no

- category, syntactic, *see* syntactic category
- Chang, C.-L., 579
- checked run-time error, 25
- Chez Scheme, 168
- `choose-move`, Prolog predicate, 563
- Clark, K.L., 579
- `class` primitive
  - in Smalltalk-80, 503
- *clause*, 528
- *clause-or-query*, 528
- client
  - versus inheritance, 417
- Clinger, W., 151
- Clocksin, W.F., 579
- close
  - over free type variables, 299
- closing
  - over free type variables, 299
- Clu, §343
  - cluster, 349, 378
  - full
    - iterators, 416
  - goal of program correctness, 377
  - mutable types, 363, 366
    - vs. immutable types, 366
  - polymorphism, §373, 376
  - real, §371
    - arrays in, 362
    - exception handling, 376
    - export lists, 352
    - iterators, 374, 378
    - syntax, §371
    - type checking, §375
  - recursively-defined data in, 362
  - relation to Pascal, 346
  - relation to Smalltalk, 397, 398, 403
  - scope rules, §359
  - semantics, §359
  - syntax, §358
  - values, 349
- Cobol, 6
- *coeff*, real Clu function, 372
- Cohen, J., 205, 579
- Cohen, N.H., 379
- Colmerauer, A., 524, 579
- Common Lisp, 63, 151, 505
- CommonLoops (object-oriented extension of Lisp), 505
- constructor
  - pitfalls as a technical term, 76
- continuation, 106
- continuation-passing style
  - in Smalltalk, 398, 414
- core language
  - in μScheme, 88
- Cox, B., 505
- *create*, real Clu function, 372
- creators, 76
  - in Smalltalk, 401
- cross-reference
  - explanation of, 23

- Dahl, O.-J., 379
- data abstraction, §347, 378
- data-type specification, 378
- De Morgan's laws
  - for `exists?` and `all?`, 100
  - in solving Boolean formulas, 163
- *declaration*, 358
- *def*, 8, 234, 281, 282, 358, 408, 521
- *degree*, real Clu function, 372
- DeGroot, D., 579
- delegation
  - in the implementation of `Set`, 451
- design patterns
  - factory, 513
- Deutsch, L.P., 188, 205, 522
- `diff`
  - Clu function, 370
- *diff*, full Clu function, 372
- `diffappend`, Prolog predicate, 550
- `different`, Prolog predicate, 562
- dispatch
  - of methods in Smalltalk, 403
- Dybvig, R.K., 151
- dynamic scoping, 119

- Eiffel (object-oriented language), 505, 514
- `either`
  - alternative elimination form for sum types, 254
- elaboration, 271
- *elements*, Clu iterator, 375
- `emptyset`
  - Scheme function, 159
- `enclosed-area`, Clu function, 355
- equality type
  - in Standard ML, 318
- equational reasoning, 77
- execute

For COMP 105, Tufts University, Spring 2014 only --- do no

as a synonym for evaluate, 397
exp, 8, 234, 358, 408
exploding heads, 96, 503
exports, 358

`fac`
Clu function, 392
factory
design pattern, 513
Felleisen, Matthias, 151
Fenichel, R., 205
FinancialHistory, real Smalltalk-10 class, 496
first-class functions
vs first-class, nested functions, 88
first-class values, 88
first-order functions, 89
Flavors (object-oriented extension of Lisp), 505
Floyd, R., 378
formals, 8, 234, 358, 408, 722
Fortran, 6
free variable, 225
Friedman, Dan, 151
function, 8, 234, 358
functional abstraction, 21
functions
first-order, 89
higher-order, 89
functor, 528
Futatsugi, K., 379

Gallier, J., 579
generality
of types, 293
generalization
of types in μML, 307
generativity
in Clu, 344
generic
meaning defined by recursion over types, 101
meaning parametrically polymorphic, 101
goal, 528
goals, 528
Goguen, J., 379
Goldberg, A., 398, 399, 496, 505, 506
Graham, Paul, 151
Gries, D., 379
ground type, 291
Guttag, J., 367, 379

Haskell, §341
Henderson, P., 151
Hewitt, C., 205
higher-order functions, 89
Hoare, C.A.R., 378, 379
Hogger, C.J., 579
Hopcroft, J., 722
Hope (functional language), 151

identifier cross-reference
explanation of, 23
Impcore
interpreter for, §23
relation to C, 11, 55
scope rules, 55
semantics, §9
syntax, §8
imperative features, 286
imperative programming, 6
impl, 358
induction step
in a recursive algorithm, 68
instance
relation between types, 293
instantiation
of an ML type scheme, 293
integer, 113, 257, 285
Iswim (functional language), 151
`it`
automatic binding in Impcore, 44
automatic binding in μScheme, 134

Johnson, R., 514
Jones, C., 379

Kaehler, T., 505
Kamin, S., 379
Kay, A., 397
Kelsey, R., 151
Knuth, D.E., 92, 205
Kowalski, R., 524, 579

lambda-bound variables, 299
λ-calculus, 524
inspiration for Lisp, 63
Landin, P.J., 151
Lee, R.C.-T., 579
`length`
Clu function, 361
let-bound variables, 299
let-keyword, 113, 257, 285

For COMP 105, Tufts University, Spring 2014 only --- do no

letrec
  less polymorphic than expected, 300
lexical scoping, 118
Liebermann, H., 205
lies, damn lies, and the logical interpretation of Prolog, 538
Lindstrom, G., 579
linear congruential method for generating random numbers, 92
Liskov, B., 367, 376, 377, 379
Lisp, §62
  inspired by λ-calculus, 63
  object-oriented extensions of, 505
  recursion in, 64, 66
`list`, real Lisp function, 68
literal, 66, 113, 257, 285, 408
Lloyd, J.W., 579
locals, 408
locations
  mutable
    in Scheme closures, 91
Loeckx, J.J.C., 379
logic programming, 578

Maier, D., 579
Manna, Z., 379, 579
mathematical logic, 524
  as a programming language, §525
  conjunctive normal form, 569, 572, 577
  first-order predicate logic, §571, 577
  Horn clause logic, 571, 578
  mechanical theorem proving, 578
  propositional logic, §567, 578
  relation to Prolog, §566
  resolution, 569, 573, 578
McCarthy, J., 63, 64, 141, 205
McCarthy, John, 151
McGettrick, A.D., 378, 379
Mellish, C.S., 579
`member`, Prolog predicate, 547
`member-also`, Prolog predicate, 547
`member?`
  Scheme function, 159
memory management, §173
  copying garbage collection, §189
  mark-and-sweep garbage collection, §183
  reference counting, §200
message categories
  in Smalltalk-80, 498
message pattern
  in Smalltalk-80, 498
metaclasses
  in full Smalltalk-80, 503
method dispatch, 403
method-definition, 408
Meyer, B., 505, 514
mini-index
  explanation of, 23
Minsky, M., 205
mixed arithmetic
  in μSmalltalk, 425
mode-change, 528
Modula-2, 6, 346, 378, 379
monomorphic, 235, 249
monomorphic function
  defined, 100
monotype, 291
Moon, D.A., 505
more general
  used to relate two types, 293
mutability
  of queues, 437
mutators
  in Smalltalk, 401

`naive-sort`, Prolog predicate, 548
Naur, P., 64
`nearer-origin`, Clu function, 352
Nilsson, N.J., 579
nonlocal return
  in Smalltalk, 449
`not_member`, Prolog predicate, 562

Objective C (object-oriented language), 505
observers, 76
  in Smalltalk, 401
occurs check
  in ML type inference, 316
  in Prolog, 535
`ordered`, Prolog predicate, 548
overloading, 101, 504
Owl (object-oriented language), 505
"own" variables, §92

$\mathcal{P}$
  as primitive function in identifier cross-reference, 23
`pair?`
  Scheme primitive, 76
parameters

For COMP 105, Tufts University, Spring 2014 only --- do no

accumulating
method of, 71
parametric polymorphism, 101
Parnas, D., 378, 379
partial application
of a curried function, 95
`partition`, Prolog predicate, 549
Pascal
as imperative language, 6
overloading, 504
relation to Clu, 346
type-checking, 371
path, 379
path-parameter, 380
Patterson, D., 505
Payne, J., 506
Perlis, A., 64
`permutation`, Prolog predicate, 548
phase distinction, 246
Poisson distribition
in discrete-event simulation, 432
Poly, real Clu cluster, 372, 374
polymorphic, 249
polymorphic function
defined, 100
polymorphism, 101, 504
ad hoc, 101
parametric, 101
subtype, 101
three kinds of, 101
polytype, 291
Pop-2 (functional language), 151
predicate, 528
primitive, 9, 113, 234, 285
program verification
abstraction function, 378
Floyd-Hoare proof rules, 378
homomorphism (abstraction function), 378
inductive assertion, 378
loop invariant, 378
representation invariant, 378
Prolog, 2, §524, 524
argument, 526
as artificial intelligence language, 577
as parallel-processing language, 579
best-known logic programming language, 524, 577
body
of rule, 526
clause, 526
closed-world assumption, 545, 577
conclusion
of rule, 526
difference lists, 550, 577
encourages code re-use, 547
fact, 526
head
of rule, 526
logical interpretation, §530
logical variable, 526, 578
occurs check, 564, 578
pattern-matching, 545
predicate, 526
premises
of rule, 526
primitive predicates, §543
procedural interpretation, §536
differs from logical interpretation, 538, 547, 566, 575
real, §564
the cut, 577
negation in, 565
semantics, §564
syntax, §564
relation to Scheme, 545
relation to mathematical logic, §566
resolution as basis for, 570, 573, 576
running programs "backwards", 531
semantics, §530
syntax, §527
unification, 535, 578
proofs
by calculation, 77
proper tail recursion, 148
property list, 74
pseudorandom numbers
generated using the linear congruential method, 92

`quadrant`, Clu function, 352
query, 528
`quicksort`, Prolog predicate, 549

random numbers
generated using the linear congruential method, 92
`recfac`, Clu function, 392
recursion, 11, 12
avoided with difference lists, 550

For COMP 105, Tufts University, Spring 2014 only --- do no

Rees, J., 151
reflection
  in Smalltalk, 409
  in Smalltalk-80, 502
Reingold, E.M., 151
returns, 358
Reynolds, R., 379
Ringwood, G.A., 579
Roberts, E.S., 151
Robinson, J.A., 579
Robson, D., 398, 399, 496, 505, 506
Rohl, J.S., 151
`rotate`, Clu function, 352
Rowe, N.C., 579

S-exp, 66, 113, 257, 285
Schaffert, C., 377, 505
Scheme
  continuation, 150
  data abstraction in, 379
  dynamically-bound (fluid) variables, 164
  own variables implemented in, 92
  polymorphism, §373
  recursion in, 65
  relation to Prolog, 545
  semantics, §65
  Smalltalk blocks related, 413
  values (S-expressions), §65
Schiffman, A.M., 522
Schmucker, K., 505
Schorr, H., 188, 205
Schriber, T., 506
scoping
  dynamic, 119
  lexical, 118
  static, 118
seed
  of a pseudorandom-number generator, 92
sentinel
  in μSmalltalk lists, 454
sets, §72
S-expressions, 65
Shapiro, E., 559, 577, 579
shared mutable state
  in a resettable counter, 91
side effect, 7
side effects, 9, 248
Siekmann, J., 579
signals, 358
Simula 67, 346, 378, 379, 506
  inspiration for Smalltalk, 397, 503, 505
Slade, S., 151
Smalltalk
  class, 398
  instance variable, 398
  message, 398
  method, 398
  mutable types, 401
  object, 398
  protocol, 398
  terminology (compared to Clu), 398
Smalltalk, 2, 378, §396
  abstract class, 416, 417, 421, 446, 454, 501, 503
  answer, 397
  class, 397, 503
  full, §496
    binary messages, 499
    blocks, 503
    class hierarchy, 398, 416, 501
    class variables, 500, 503
    Collection hierarchy, §502
    everything an object, 504
    literals, 500
    numbers, §501
    precedence, 499
    semantics, §500
    syntax, 400, §496, 504
    `UndefinedObject` class, §445
  inheritance, 504
    multiple, 504
  instance variable, 402, 403, 410, 504
  message, 397, 410, 504
  method, 397, 410, 504
  method dispatch, 411
  method search, 411, 445, 504, 522
  mutable types, 404, 416, 422, 451
  object, 397, 504
  overloading, 504
  polymorphism, 419, 504
  protocol, 397, 504
  receiver, 397
  relation to Clu, 397, 398, 403
  `self`, 410, 504
  Simula 67 inspiration for, 397, 503, 505
  simulation in, §427
  subclass, 397, 504
  super, 411, 504
  `super`, 410

For COMP 105, Tufts University, Spring 2014 only --- do no

  - superclass, 397, 504
  - syntax, §407
  - terminology (compared to Clu), 397
  - type checking difficult in, 514
- μSmalltalk
  - arity of message names, 408
- Snyder, A., 376, 377, 379
- *sort*
  - real Clu function, 376
- sorting, §72
- soundness
  - as applied to inference rules, 303
- soundness proof
  - for inference rules using explicit substitutions, 303
- `sqrdist`, Clu function, 352
- Standish, T.A., 205
- static scoping, 118
- Steele, G.L., Jr., 151, 164
- Steele, Guy L., Jr., 63, 151
- Sterling, L., 559, 577, 579
- streams
  - used in μScheme to read definitions from a file, 222
- subtype polymorphism, 101
- *sugared-function*, 358
- `suggest`, Prolog predicate, 563
- super
  - messages to, 410
- suspension, 645
- Sussman, G.J., 151, 164, 379
- Sussman, Gerald J., 63
- syntactic category, 7
  - precise definition of, 53
- syntax, context-free, §721

- T (Scheme dialect), 151
- Tarnlund, S.A., 579
- *term*, 528
- thunk, 645
- *toplevel*, 721
- Touretzky, D., 151
- tracing
  - Smalltalk, 491
- `transform`, Prolog predicate, 562
- `transform2`, Prolog predicate, 563
- tree traversal, §83, 183, 188, 205
- *type*, 234
- type scheme, 291
- type variable
  - key to understanding thereof, 260
- *type-exp*, 257, 358
- *type-formals*, 358
- `type-lambda`
  - in Typed μScheme, 257, 262
- *types*, 358
- *tyvar*, 379

- Ullman, J., 722
- unchecked run-time error, 25
  - from using an unspecified value, 122
- unfold, 648
- Ungar, D., 205
- unifier, 301
- unspecified
  - in μScheme semantics, 122
- unspecified values
  - preventing misuse by careless persons, 605
- *use*, 528

- *value*, 8, 234, 358
- value restriction, 714
- *variable*, 528

- Waite, W., 188, 205
- Warren, D.S., 579
- Wexelblat, R., 64
- Wilensky, R., 151
- Winograd, T., 579
- Winston, P.H., 579
- Winston, Patrick H., 151
- Wirth, N., 379
- Wos, L., 579
- Wrightson, G., 579
- Wulf, W., 379

- Yochelsen, J., 205

- Zilles, S., 379

For COMP 105, Tufts University, Spring 2014 only --- do no