

References

1. Abadi, M., Cardelli, L.: *A Theory of Objects*. Springer, Berlin (1998)
2. Dybvig, R.K.: *The Scheme Programming Language*, 2nd edn. Prentice Hall, New York (1996). www.scheme.com/tspl2d/
3. Gunter, C.A.: *Semantics of Programming Languages: Structures and Techniques*. MIT Press, Cambridge (1992)
4. Kahn, G.: Natural semantics. In: *Proceedings of the Symp. on Theoretical Aspects of Computer Science, TACS, Passau* (1987)
5. Mitchell, J.C.: *Foundations for Programming Languages*. MIT Press, Cambridge (1996)
6. Mitchell, J.C.: *Concepts in Programming Languages*. Cambridge University Press, Cambridge (2002)
7. Peyton Jones, S., Lester, D.: *Types and Programming Languages*. Prentice Hall, New York (1992)
8. Pierce, B.C.: *Types and Programming Languages*. MIT Press, Cambridge (2002)
9. Plotkin, G.D.: LCF considered as a programming language. *Theor. Comput. Sci.* **5**, 223–255 (1977)
10. Plotkin, G.D.: A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, Aarhus, Denmark, September 1981
11. Reynolds, J.C.: *Theories of Programming Languages*. Cambridge University Press, Cambridge (1998)
12. Scott, D.: Continuous Lattices. *Lecture Notes in Math.*, vol. 274, pp. 97–136. Springer, Berlin (1972)
13. Weis, P., Leroy, X.: *Le langage Caml*, 2nd edn. Dunod, Paris (1999)
14. Winskel, G.: *The Formal Semantics of Programming Languages*. MIT Press, Cambridge (1993)

Index

A

Abstract machine, 43
Algorithm
 Damas and Milner, 70
 Hindley's, 64
 Robinson's, 66
 α -equivalence, 11
Alphabetic equivalence, 11
Arity, 7
Array, 81

B

β -reduction, 19
Binding
 dynamic, 23
 late, 88
 static, 23

C

Call by name, 26, 28, 33
Call by value, 27, 29, 35
Church numeral, 20
Closed set, 4
Closure, 34
 recursive, 38
Compiler, 43
 bootstrapping, 43
Composition, 12
Confluence, 24
Constant, 7
Continuous function, 2

D

De Bruijn index, 36
Definition
 explicit, 1
 inductive, 4

Derivation, 5
Deterministic, vi

E

Environment, 33
 semantic, 56
 typing, 53
Evaluate, 22
Evaluator, 25

F

Fields, 81
Fixed point
 Curry, 23
 first theorem, 2
 function construction via, 38
 in PCF, 17
 second theorem, 3
Functionalisation, 78

H

Height, 10

I

Interpreter, 33

L

Label, 82
Language, 7
Limit, 1
List, 42

M

Method, 85
 dynamic, 86
 static, 86
Module, 81

N

Number of arguments, 7

O

Object, 85

Ordering, 1

 Scott's, 58

 strongly complete, 2

 weakly complete, 1

P

Pair, 42, 81

PCF (*Programming language for computable functions*), 15

Polymorphism, 68

Position numerals, 20

R

Record, 82

 in call by name, 83

 in call by value, 83

Redex, 19

Reduction

 call by name, 26

 call by value, 27

 lazy, 27

 weak, 26

Reference, 73

Register, 44

 accumulator, 44

 code, 45

 environment, 45

 stack, 44

Renaming, 67

Result, 22

Rule, 4

S

Semantics

 big-step operational, 12

 denotational, 12

 small-step operational, 12

Side effect, 74

Solution, 65

 principal, 66

Sort, 9

Strategy, 25

Subject reduction, 55

Substitution, 10

System

 F, 71

 T, 61

T

Term, 7

 closed, 10

 irreducible, 22

 stuck, 22

Thunk, 33

Tree, 42

Type, 52

 checking, 54

 inference, 63

 principal, 66

Type preservation

 by interpretation, 56

Type scheme, 68

U

Unification, 66

V

Value, 22, 26

 extended, 35

 rational, 40

Variable, 8

 capture, 11

 environment, 36

 mutable, 79