

## Index

---

- Abella, 228
- abstract datatypes, 165
- abstract logic programming language, 73
- abstraction
  - body of, 97
  - bound variable of, 97
- `accum_sig` keyword, 156, 285
- `accumulate` keyword, 154
- $\forall\exists$  unification problems, 32
- $\forall\exists\forall$  unification problems, 106, 212
- algorithm W, 260
- $\alpha$ -conversion, 37, 100
- $\alpha$ -rewriting, 100
- anonymous variable, 47, 51
- answer substitution, 43, 52
- `append` predicate, 50
- application of a term, 17
- argument type, 13
- `assert`, Prolog predicate, 287
- atomic formula, first-order, 34
  
- backchaining, 3, 44
- backslash `\`
  - as  $\lambda$ -binder, 97
  - as quantifier binder, 36
- `bagof`, 190
- Bedwyr, 228
- $\beta$ -conversion, 6, 7, 100
- $\beta$ -expansion, 100
- $\beta$ -normal form, 101, 185
  - arguments of, 111
  - binder of, 111
  - flexible, 111
  - head of, 111
  - rigid, 111
- $\beta$ -redex, 101
- $\beta$ -reduction, 2, 100
  
- $\beta_0$ -conversion, 206, 220
- $\beta\eta$ -long normal form, 111
- $\beta\eta$ -normal form, 102
- big-step specification, 250
- binary clauses, 130, 148
- binder mobility, *see* mobility of binders
- bisimulation, 271
- `bnorm`,  $\beta$ -normal form, 186
- bound variable, 36
- bound variable renaming, 37
  
- Calculus of Constructions, 148
- call-by-name evaluation, 182, 273
- call-by-value evaluation, 182
- candidats de réductibilité, 5
- capture avoiding substitution, 86, 183
- Church numerals, 102, 115
- Church, Alonzo, 6, 74, 116, 147, 208
- classical logic, 4, 41, 235
- classical logic theorem prover, 239
- classical vs intuitionistic logic, 89
- clausal order, 76
- `close_in`, built-in predicate, 287
- `close_out`, built-in predicate, 287
- closed formula, 38
- closed term, 98
- closed-world assumption, 94
- comparisons `<`, `>`, `=<`, `>=`, 49
- complete set of unifiers, 213
- complete trace, 269
- `compose`, 130
- comprehension, 5
- computation via proof normalization, 2
- computation via proof search, 2
- computation-as-deduction, 2
- computation-as-model, 2
- concrete nonsense, 183

302

conditional, 137, 251  
 consistent, 119  
 constant symbols, 14  
 context, 17  
 continuation passing style, 131, 256  
   administrative redexes, 257  
   Fischer CPS transformation, 257  
 contraction rule, 245  
 copy-clauses, 199, 210  
 cut admissibility, 63  
 cut rule, 2, 63, 72  
 cut, Prolog pruning operator  $!$ , 81, 137, 144, 240, 287  
 cut-elimination theorem, 3, 63, 72, 148  
  
 de Bruijn numerals, 192, 209  
 definite formulas, 42  
 definition by type lowering, 201, 202  
 denotational semantics, 258  
 dependently typed  $\lambda$ -calculus, 33, 210  
 determinate type expressions, 67  
 difference lists, 140  
   as functions, 140  
 Diophantine equations, 115  
  
 eigenvariable, 82  
 empty types, 108  
 equality  
   extensional, 146  
   intensional, 146  
   Leibniz's, 120  
 essentially existential, 205  
 essentially universal, 205  
 $\eta$ -contraction, 101  
 $\eta$ -conversion, 6, 7, 101  
 $\eta$ -expansion, 101  
 $\eta$ -redex, 101  
 $\eta$ -reduction, 101  
 excluded middle principle, 42  
 existential generalization, 203  
 existential quantifier as  $\sigma$ , 36  
 explicit substitutions, 209  
`exportdef` keyword, 284  
 extensional equality, 146  
 extensions of functional expressions, 146  
  
`fail`, built-in predicate, 287  
 Fibonacci numbers, 132, 248  
   `fib_memo`, 132  
 finite state machines, 53  
 first-order  $\Sigma$ -formula, typed, 38  
 first-order  $\Sigma$ -term, typed, 18  
 first-order hereditary Harrop formulas, 76

Index

first-order Horn clauses, 34, 43  
 first-order terms, typed, 14  
 first-order unification, 33  
 Fischer CPS transformation, 257  
 fixity, 16  
 flexible atom, 119  
   as goal formula, 132  
   as head of clause, 119  
 flexible term, 111, 214  
 flexible-flexible equations, 113  
 flexible-flexible unification problems,  
   113, 217  
 flexible-rigid equations, 113  
 focused proofs, 73  
`fohc`, 43, 63  
`fohh`, 76  
`fold1`, 130  
 $\forall$ -lifting, 227  
`foreach`, 126  
 formula, first-order, 38  
 formula, higher-order, 99  
`forsome`, 126  
 free for, 85, 100  
 function symbols, 14  
 functional and logic programming, 148  
 functional difference lists, 140, 149, 238  
 functional programming, 2  
 functional type, 13  
  
 general models, 147  
 generic judgment, 94  
 generic judgments, 83  
 generic vs universal judgment, 94  
 Gentzen, Gerhard, 3, 72, 245  
 Girard, Jean-Yves, 5, 73  
 goal formulas, 38  
 goal-directed proof search, 39, 73  
`goalreduce`, 241  
 Gödel programming language, 32, 173  
 Gödel, Kurt, 5  
 $>$ , built-in predicate, 49  
 $>=$ , built-in predicate, 49  
  
`halt`, built-in predicate, 287  
 Harrop formulas, 93  
 Herbrand universe, 123  
   for `hohc`,  $\mathcal{H}_1^\Sigma$ , 122, 148  
   for `hohh`,  $\mathcal{H}_2^\Sigma$ , 122  
 Herbrand's theorem, 147  
 hereditary Harrop formulas, 4, 8, 93  
   first-order, 76  
 higher-order abstract syntax, 204, 209

- higher-order hereditary Harrop formulas, 122, 124
  - extended,  $hohh^+$ , 122, 125
- higher-order Horn clauses, 122
- higher-order logic, 5, 147
  - cut elimination, 5
  - mathematical axioms, 6
  - predicate quantification, 5, 6
  - theorem proving, 226
- higher-order logic programming, 8
  - examples, 126
- higher-order matching, 227
- higher-order pattern unification, 9, 210, 220, 227, 288
  - complexity, 227
  - decidability, 222
  - most general unifiers, 222
  - occurs-check, 223
  - variable elimination, 223
- higher-order programming in functional programming, 146
- higher-order unification, 5, 9, 96, 111, 143, 210, 211, 288
  - checking unifiability, 214, 216
  - complete set of unifiers, 213
  - flexible-flexible problems, 217
  - imitation substitutions, 215
  - lack of most general unifiers, 213
  - matching tree, 219
  - non-terminating reductions, 218
  - pre-unification, 218, 219, 228
  - projection substitutions, 215
  - redundant search, 226
  - simplification, 214
  - undecidability, 214, 220, 226
  - via combinators, 227
  - via explicit substitutions, 227
- Hilbert's Tenth Problem, 115
- HiLog, 149
- $hohc$ , 122, 147
- $hohh$ , 122, 124, 147
- $hohh^+$ , 122, 125, 160, 161
- Horn clauses, 4, 8
- Huet, Gerard, 116
- $if$ , 137
- imitation substitutions, 215
- implicational goals, 77
- implicational goals in Teyjus, 282, 287
- $in\_stream$ , primitive type, 10, 287
- incomplete proof strategy, 60
- incompleteness of second-order logic, 5
- incompleteness theorem, 5
- inconsistent, 90, 119
- inference rule, completeness, 41
- inference rule, soundness, 41
- $infix$ ,  $infixl$ ,  $infixr$ , 16
- input, built-in predicate, 287
- $int$ , primitive type, 10, 15, 286
- $int\_to\_string$ , built-in predicate, 286
- integer constants, 15
- intensional equality, 146
- intensions of functional expressions, 146
- intuitionistic logic, 4, 41, 245
  - disjunctive property, 93
  - existential property, 93
- intuitionistic vs minimal logic, 89
- invertible rules, 243
- $is$ , built-in predicate, 67, 132, 193, 238, 286
- Isabelle, 227
- kind keyword, 12
- kinds, 11
  - declarations, 12
  - expressions, 11
- Kripke model, 94
- $L_\lambda$ , 9, 205, 210, 214, 227
  - defining condition, 220
  - dynamic vs static, 226, 228
- $L_\lambda$ -unification, *see* higher-order pattern unification
- $\lambda$ -conversion, 5
- $\lambda$ Prolog, 7
- $\lambda$ -conversion, 100, 101
- $\lambda$ -normal form, 101
  - of  $s$ ,  $\lambda norm(s)$ , 102
- $\lambda$ -terms, 6, 97
  - substitution application, 212
  - untyped, 178
  - well-formed over a signature, 98
- $\lambda$ -tree syntax, 204, 209, 262
- left-introduction rules, 42, 45
- Leibniz's equality, 120
- $<$ , built-in predicate, 49
- $=<$ , built-in predicate, 49
- LF, dependently typed  $\lambda$ -calculus, 33, 148, 246
- linear logic, 4, 73
- $list$ , type constructor, 11, 15
- $::$ , list constructor, 15
- $nil$ , list constructor, 15
- logic program, 39
- logic programming, 1
  - computational dynamics, 4
  - expressivity, 3
  - logical primitives, 3

- logic variable, 61
- logic variables in clauses, 86
- logical constants, 35
- logics
  - classical, 4, 89, 235
  - intuitionistic, 4, 89, 245
  - linear, 4, 73
  - minimal, 89
- .lp, output of `tjlink`, 281
- .lpo, output of `tjcc`, 281
  
- mapfun, 139
- mappred, 126, 139
- matching tree, 219
  - completeness property, 219
  - infinite paths, 219
- mathematical proofs, 2
- matrix of a unification problem, 211
- may judgments, 269
- mechanization of reasoning
  - cut rule, 3
  - predicate quantification, 6
- memb\_and\_rest, 152
- meta-level programming, 73
- miniFP
  - big-step specifications, 251
  - call-by-value evaluation, 251
  - continuation passing style, 256
  - evaluation using contexts, 253
  - let-polymorphism, 260
  - polymorphic types, 250
  - type inference, 249
  - unfolding, 256
- minimal logic, 89
- minimal logic negation, 90
- Minlog, 228
- mixed evaluation, 256
- mixed quantifier prefix, 105, 211
- ML programming language, 21, 24, 32, 74
- mobility of binders, 184, 185, 199, 205
- .mod, file extension for modules, 280
- module accumulation, 154
- module elaboration, 164, 172
- module inlining, 172
- module keyword, 152
- modules, 50
- most general unifier, 28, 208, 213
- multiple solutions, 53
- multiset, 26
- must judgments, 269
  
- $\nabla$ -quantifier, 94, 275
- nameless dummies, 192, 209
  
- natural deduction, 231
- natural join, 130
- negation in logic programming, 94
- negation normal form, 195, 236
- negation-as-failure, 137, 269, 272
- negative subformula occurrence, 76
- nominal logic, 210
- non-standard models, 5
- non-termination, 51, 60
- nonfunctional type, 13
- not, built-in predicate, 137, 287
  
- o, primitive type, 10, 15, 286
- O-proof, 45, 73, 77, 89
- object-level substitution
  - using  $\beta$ -reduction, 180, 199
  - using copy-clauses, 199
- occurs-check, 30, 32, 33, 223
- open term, 98
- open\_in, built-in predicate, 287
- open\_out, built-in predicate, 287
- open-world assumption, 94
- operator declarations, 16
- order of a type expression, 13
- out\_stream, primitive type, 10, 287
- output, built-in predicate, 287
  
- pair, type constructor, 11, 16
- palindrome, 142
- parameterized modules, 168
- parametric polymorphism, 20
- Parinati, 246
- partial evaluation, 255
- paths in  $\lambda$ -terms, 188
- pattern unification (higher-order), *see*
  - higher-order pattern unification
- Peirce's formula, 89
- percent symbol (%) introduces comment, 23
- pervasive constants, 14, 153
- pervasive predicates, 49
- pervasive signature-program pair, 49
- pervasive types, 14, 153
- $\pi$  as universal quantifier, 99
- $\pi$ -calculus, 95, 261
  - encoding evaluation, 273
  - free and bound actions, 264
  - input action (`dn x y`), 263
  - internal mobility  $\pi_i$ , 275
  - labeled transitions, 263, 266
  - output action (`up x y`), 263
  - silent action  $\tau$ , 263
  - simulation and bisimulation, 271
  - traces, 267

- polarity of logical symbols, 121
- polymorphic constants, 15
- polymorphic typing, 12, 15, 67
- positive subformula occurrence, 76
- Post correspondence problem, 114, 214
- `postfix`, `postfix1`, 16
- pre-unification, 218, 219, 228
- precedence of operators, 16
- predicate quantification, 5
- predicate symbol, 34
- `prefix`, `prefixr`, 16
- prenex normal form, 195
- primitive types, 13
- `print`, built-in predicate, 49
- program clauses, 38
- program clauses, with existential quantification, 160
- programming-in-the-large, 150
- projection substitutions, 215
- Prolog, 7
- Prolog/Mali implementation of  $\lambda$ Prolog, 32
- proof objects-as-terms, 233
- propositional intuitionistic logic
  - decision procedure, 231
  - PSPACE-complete, 245
- propositional symbols, 35
- pruning substitution, 223, 224
  
- quantifier prefix
  - $\forall\exists$ , 32
  - $\forall\exists\forall$ , 106, 212
  - simplification using raising, 108
  - simplification using Skolemization, 108
- quantifier-free formula, first-order, 35
- queries, 39
  
- raising, 108, 208, 212, 227
  - delayed application, 225, 228
- reachability problem, 53
- `read`, built-in predicate, 49
- read-prove-print loop, 50
- real constants, 15
- `real`, primitive type, 10, 15, 286
- recursively defined types, 15, 19
- `red1`, 186, 241, 258
- `redex`, 186, 241
- `reduce`, 186
- `reducefun`, 139
- reflexive closure, `ref`, 127
- relational composition, 130, 243
- representation independence, 151
- resolution, 147
- resolution refutation, 72
  
- `retract`, Prolog predicate, 287
- `reverse`
  - correctness proof, 134
  - in `fohh`, 86
  - in `hohh`<sup>+</sup>, 125
  - using higher-order relations, 130
- right-introduction rules, 40, 45
- rigid atom, 119, 161
- rigid term, 28, 111, 214
- rigid-flexible equations, 113
- rigid-rigid equations, 113
  
- scope extrusion, 89, 95
- scoped constants, 82
- search semantics of connectives, 39
- second-order logic, 5
- separate compilation, 151, 172
- sequent, 39
- sequent calculus, 2, 39, 72
- `.sig`, file extension for signature, 280
- `sig` keyword, 153
- $\sigma$  as existential quantifier, 99
- $\Sigma$ -formula, first-order, typed, 38
- $\Sigma$ -term
  - first-order, typed, 18
  - higher-order, typed, 98
- signature, 17, 38, 153
- signature accumulation, 155
- signature dependent definition, 202
- signature elaboration, 156
- signature matching, 153, 158
- signature merging, 156
- signature-program pair,  $(\Sigma, \mathcal{P})$ , 39
- Simple Theory of Types, 6, 8, 74, 116, 147, 208
- simply typed  $\lambda$ -calculus, 96
- simply typed  $\lambda$ -terms, 98
- simulation, 271
- size of a  $\lambda$ -term, 103
- Skolemization, 72, 108, 147
- small-step specifications, 251, 263
- solution to a unification problem, 109
- sorts, 10
- Standard ML, 173
- standard model of arithmetic, 5
- `std_err`, primitive type, 15, 287
- `std_in`, primitive type, 15, 287
- `std_out`, primitive type, 15, 287
- sterile jar, 82
- string constants, 15
- `string`, primitive type, 10, 15, 286
- strong typing, 68
- structural operational semantics, 259

subject-reduction theorem, 191  
 sublist, 126  
 subst predicate, 199, 225  
 substitution in quantified formulas, 84  
 substitution lemmas for free, 186, 192, 198  
 substitution of  $t$  for  $x$  in  $s$ ,  $s[t/x]$ , 85, 100, 101  
 suspension calculus, 209  
 symmetric closure, *sym*, 127

tactic, 241  
 tacticals, 243  
 target type, 13  
 term reduction, 28  
 term rewriting, 145  
 terms  
   application, 17  
   constant symbols, 14  
   first-order, typed, 14  
   variables, 17

Teyjus, 7, 9, 170, 174, 228, 277  
   compiler, *tjcc*, 278  
   dependency analyzer, *tjdepend*, 278  
   disassembler, *tjdis*, 278  
   emulator, *tjsim*, 278  
   linker, 278  
   linker, *tjlink*, 278  
   loader, 277

theorem prover  
   first-order classical logic, 239  
   propositional intuitionistic logic, 231

tied variables in clauses, 87  
*tjcc*, 278, 280  
*tjdepend*, 278, 285  
*tjdis*, 278  
*tjlink*, 278  
*tjsim*, 278  
 trace equivalence, 271  
 transitive closure, *trans*, 127  
 transparent type variable, 67  
 Twelf, 33, 148, 228  
 type checking, 68  
 type, type expression classifier, 11  
 type constructors, 11  
 type declarations, 16  
 type expressions, 12  
 type inference, 18, 38, 68, 74, 98, 190, 279  
 type keyword, 16, 67  
 type preservation theorem, 191  
 typed first-order unification, 26  
 types

constructed types, 12  
 functional and nonfunctional, 13  
 order, 13  
 primitive types, 13  
 role in unification, 31, 224, 228  
 type expressions, 12  
 variables, 12

typing judgments  
 $\sigma \triangleleft_f \tau$ , 18  
 $\Sigma; \Gamma \vdash_f t: \tau$ , 18  
 $\Sigma; \Gamma \vdash t: \tau$ , 97

$\_.$ , anonymous variable, 47, 51

unification, 61  
   first-order, typed, 26  
   higher-order, 210, 211  
   higher-order pattern, 210  
   of  $\lambda$ -terms, 211  
   term reduction, 28  
   variable elimination, 28

unification problems, 26, 105  
   as quantified equalities, 32, 105  
   first-order, 33  
   flexible-flexible, 113  
   solutions, 109  
   unifiers, 109

unification under a mixed prefix, 105, 227  
 unifier, 26, 109  
 uniform proofs, 73  
 union, 130  
 universal judgment, 94  
 universal quantifier  
   extensional reading, 83, 94  
   intensional reading, 83, 94  
 universal quantifier as *pi*, 36  
 untyped  $\lambda$ -terms, 178  
    $\beta$ -normal form, 186  
   paths within, 188  
   semantics, 187

*use\_sig* keyword, 285  
*useonly* keyword, 285

vacuous, 195  
 variable capture, 85  
 variable elimination, 28, 223  
 variables, term, 17  
 variables, type, 12

Warren Abstract Machine, 74