

Bibliography

- [1] J. Backus. Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. *CACM*, 21(8):613–641, August 1978.
- [2] F. W. Burton and R. L. Page. Distributed random number generation. *Journal of Functional Programming*, 2(2), 203–212, April 1992.
- [3] D. G. Carta. Two fast implementations of the minimal standard random number generator. *CACM*, 33(1), 87–88, January 1990.
- [4] H.B. Curry and R. Feys. *Combinatory Logic*. North-Holland Pub. Co., Amsterdam, 1958.
- [5] L. Damas and R. Milner. Principal type schemes for functional programs. In *Proceedings of the 9th ACM Symposium on Principles of Programming Languages*, pages 207–212, Albuquerque, N.M., January 1982.
- [6] K-F. Faxén. A static semantics for Haskell. *Journal of Functional Programming*, 12(4), 295–358, July 2002.
- [7] P. Hellekalek. Don't trust parallel Monte Carlo. *ACM Simulation Digest*, 28(1):82–89, July 1998.
- [8] J.R. Hindley. The principal type scheme of an object in combinatory logic. *Transactions of the American Mathematical Society*, 146:29–60, December 1969.
- [9] P. Hudak, J. Fasel, and J. Peterson. A gentle introduction to Haskell. Technical Report YALEU/DCS/RR-901, Yale University, May 1996.
- [10] Mark P. Jones. A system of constructor classes: overloading and implicit higher-order polymorphism. *Journal of Functional Programming*, 5(1), January 1995.
- [11] Mark P. Jones. Typing Haskell in Haskell. *Haskell Workshop*, Paris, October 1999.
- [12] S. K. Park and K. W. Miller. Random number generators – good ones are hard to find. *CACM*, 31(10), 1192–1201, October 1988.

- [13] P. Penfield, Jr. Principal values and branch cuts in complex APL. In *APL '81 Conference Proceedings*, pages 248–256, San Francisco, September 1981.
- [14] S.L. Peyton Jones. *The Implementation of Functional Programming Languages*. Prentice-Hall International, Englewood Cliffs, New Jersey, 1987.
- [15] Unicode Consortium. *Unicode Character Data and Mappings*. <http://unicode.org>.
- [16] P. Wadler and S. Blott. How to make *ad hoc* polymorphism less *ad hoc*. In *Proceedings of the 16th ACM Symposium on Principles of Programming Languages*, pages 60–76, Austin, Texas, January 1989.

Index

Index entries that refer to nonterminals in the Haskell syntax are shown in an *italic* font. Code entities are shown in `typewriter` font. Ordinary index entries are shown in a roman font.

- !, 47, 173, 174, 176, 177
- !!, 57, 114, 115
- \$, 57, 105, 110
- \$!, 105, 110
- %, 149, 150
- &&, 57, 81, 105, 110
- (*r*), 83
- (*r r*), 83
- (), *see* trivial type and unit expression
- *, 57, 92, 93, 105, 106
- ** , 57, 93, 105, 107
- + , 57, 92, 93, 105, 106, *see also* *n+k* pattern
- ++ , 57, 114
- , 57, 92, 93, 105, 106, *see also* negation
- ., 57, 83, 105, 110
- / , 57, 92, 93, 105, 107
- // , 173, 175–177
- /= , 57, 86, 105, 140
- :, 57, 82
- :+ , 153, 154
- :: , 32
- < , 57, 86, 105, 140
- <= , 57, 86, 105, 140
- =<< , 90, 105, 109
- == , 57, 86, 105, 140
- > , 57, 86, 105, 140
- >= , 57, 86, 105, 140
- >> , 57, 90, 99, 105, 109
- >>= , 57, 90, 99, 105, 109
- @ , *see* as-pattern
- [] (nil), 82
- ⊥ , 19, 224
- \\ , 179, 180, 185
- \ , *see* lambda abstraction
- \a , 12
- \b , 12
- \f , 12
- \n , 12
- \r , 12
- \t , 12
- \v , 12
- \& , 12
- \\ , 181
- ^ , 57, 93, 94, 105, 109
- ^^ , 57, 93, 94, 105, 109
- _ , *see* wildcard pattern
- | | , 57, 81, 105, 110
- ~ , *see* irrefutable pattern
- abbreviated module, 69
- abs, 93, 94, 106
- AbsoluteSeek, 213
- abstract datatype, 46, 78
- accum, 173, 175, 177
- accumArray, 173, 175, 177
- acos, 93, 107
- acosh, 93, 107
- addToClockTime, 226, 228
- aexp, 18, 22, 24–26, 136
- algebraic datatype, 45, 69, 139
- all, 118
- alt, 27, 136
- alts, 27, 136
- ambiguous type, 53, 64
- and, 118
- ANY, 9, 127
- any, 9, 127
- any, 118
- ANYseq, 9, 127
- ap, 199, 203
- apat, 33, 137
- appendFile, 99, 124
- AppendMode, 210
- application, 21
 - function, *see* function application
 - operator, *see* operator application
- approxRational, 94, 95, 149, 150

- arithmetic operator, 92
- arithmetic sequence, 25, 82
- array, 173
 - accumulated, 175
 - derived, 176
- Array (datatype), 174
- Array (module), 160, 173, 176, 195
- array, 173, 174, 177
- as-pattern (@), 33, 35
- ascDigit*, 9, 127
- ascii*, 12, 128
- ASCII character set, 8
- ascLarge*, 9, 127
- ascSmall*, 9, 127
- ascSymbol*, 9, 127
- asin, 93, 107
- asinh, 93, 107
- assocs, 173, 174, 177
- asTypeOf, 113
- atan, 93, 107
- atan2, 94, 95, 108
- atanh, 93, 107
- atype*, 42, 134

- basic input/output, 97
- binding, 39
 - function, *see* function binding
 - pattern, *see* pattern binding
 - simple pattern, *see* simple pattern binding
- body*, 68, 133
- Bool (datatype), 81, 110
- boolean, 81
- Bounded (class), 91, 106
 - derived instance, 53, 141
 - instance for Char, 110
- bounds, 173, 174, 177
- bracket, 207, 217
- bracket_, 207, 217
- break, 117
- btype*, 42, 134
- BufferMode (datatype), 206

- CalendarTime (datatype), 226, 228
- calendarTimeToString, 226, 228
- case expression, 27
- catch, 100, 123, 224
- catMaybes, 191, 192
- cdecl*, 40, 50, 134
- cdecls*, 40, 50, 134
- ceiling, 94, 95, 107
- changing the directory, 221
- char*, 12, 128
- Char (datatype), 82, 110
- Char (module), 119, 160, 193, 195, 227
- character, 82
 - literal syntax, 12
- character set
 - ASCII, *see* ASCII character set
 - transparent, *see* transparent character set
- charesc*, 12, 128
- chr, 193, 196
- cis, 153, 154
- class, 40, 49
- class*, 44, 134
- class assertion, 43
- class declaration, 49, 70
 - with an empty where part, 51
- class environment, 44
- class method, 40, 41, 50, 51
- clock time, 225
- ClockTime (datatype), 225, 227
- closecom*, 9, 127
- closing a file, 210
- closure, 76
- cname*, 71, 133
- cntrl*, 12, 128
- coercion, 95
- comment, 9
 - end-of-line, 9
 - nested, 9
- comment*, 9, 127
- compare, 86, 105, 140
- Complex (module), 153, 154
- con*, 20, 137
- concat, 114
- concatMap, 114
- conditional expression, 23
- conid*, 10, 11, 128
- conjugate, 153, 154
- conop*, 20, 137

- `const`, 83, 110
- `constr`, 45, 135
- `constrs`, 45, 135
- constructor class, 40
- constructor expression, 42
- `consym`, 10, 128
- context, 43
- `context`, 44, 134
- context reduction, 61
- `cos`, 93, 107
- `cosh`, 93, 107
- cosine, 95
- CPU time, 233
- `CPUTime` (module), 233
- `cpuTimePrecision`, 233
- `createDirectory`, 219, 220
- creating a file, 210
- `ctDay`, 226, 228
- `ctHour`, 226, 228
- `ctIsDST`, 226, 228
- `ctMin`, 226, 228
- `ctSec`, 225
- `ctSin`, 227
- current directory, 221
- `curry`, 83, 113
- Curry, Haskell B., vii
- `cycle`, 116
- dashes*, 9, 127
- data constructor, 45
- data declaration, 29, 45
- datatype, 45
 - abstract, *see* abstract datatype
 - algebraic, *see* algebraic datatype
 - declaration, *see* data declaration
 - recursive, *see* recursive datatype
 - renaming, *see* newtype declaration
- `dateFmt`, 231
- `dateTimeFmt`, 231
- `Day` (datatype), 225, 227
- `dclass`, 45, 135
- decimal*, 11, 128
- `decl`, 40, 58, 134
- declaration, 39
 - class, *see* class declaration
 - datatype, *see* data declaration
 - default, *see* default declaration
 - fixity, *see* fixity declaration
 - import, *see* import declaration
 - instance, *see* instance declaration
 - within a `class` declaration, 50
 - within a `let` expression, 27
 - within an `instance` declaration, 51
 - declaration group, 60
 - decls*, 40, 134
 - `decodeFloat`, 94, 95, 108
 - default class method, 50, 52, 143
 - default declaration, 53
 - `defaultTimeLocale`, 231, 232
 - `delete`, 180, 181, 185
 - `deleteBy`, 180, 183, 185
 - `deleteFirstBy`, 180, 183, 185
 - deleting directories, 220
 - deleting files, 220
 - denominator, 149, 150
 - dependency analysis, 60
 - derived instance, 53, 139, *see also* instance declaration
 - deriving*, 45, 135
 - `diffClockTimes`, 226, 228
 - digit*, 9, 127
 - `digitToInt`, 193, 195
 - directories, 208, 219
 - `Directory` (module), 219
 - `div`, 57, 92, 93, 105, 106
 - `divMod`, 93, 106
 - `do` expression, 29, 99
 - `doDiv`, 161
 - `doesDirectoryExist`, 220
 - `doesFileExist`, 220
 - `Double` (datatype), 91, 94, 112
 - `drop`, 117
 - `dropWhile`, 117
 - `e`, 166
 - `Either` (datatype), 83, 111
 - `either`, 83, 111
 - `elem`, 57, 114, 118
 - `elemIndex`, 180, 181, 185
 - `elemIndices`, 180, 181, 185

- elems, 173, 174, 177
- encodeFloat, 94, 95, 108
- end of file, 211
- entity, 67
- Enum (class), 53, 88, 106
 - derived instance, 53, 140
 - instance for Char, 110
 - instance for Double, 112
 - instance for Float, 112
 - instance for Ratio, 151
 - superclass of Integral, 106
- enumeration, 170
- enumFrom, 88, 106
- enumFromThen, 88, 106
- enumFromThenTo, 88, 106
- enumFromTo, 88, 106
- environment
 - class, *see* class environment
 - type, *see* type environment
- environment variables, 224
- EQ, 83
- Eq (class), 86, 91, 105
 - derived instance, 53, 140
 - instance for Array, 177
 - instance for Char, 110
 - superclass of Num, 106
 - superclass of Ord, 105
- error, 4, 19
- error, 19, 114, 224
- escape, 12, 128
- even, 93, 108
- exception handling, 100
- executable, 219
- execution time, 233
- ExitCode (datatype), 223
- ExitFailure, 223
- exitFailure, 223
- ExitSuccess, 223
- exitWith, 223
- exp^i , 18, 136
- exp, 18, 21–23, 27, 29, 32, 136
- exp, 93, 107
- exponent, 94, 96, 108
- exponentiation, 93
- export, 69, 133
- export list, 69
- exports, 69, 133
- expression, 4, 17
 - case, *see* case expression
 - conditional, *see* conditional expression
 - let, *see* let expression
 - simple case, *see* simple case expression
 - type, *see* type expression
 - unit, *see* unit expression
- expression type-signature, 32, 54
- expt, 161
- expts, 161
- f, 166
- fail, 90, 101, 109, 208
- False, 81
- fbind, 30, 136
- fexp, 18, 21, 136
- FFFormat (datatype), 162
- field label, *see* label, 46
 - construction, 30
 - selection, 30
 - update, 31
- fielddecl, 45, 135
- file, 208
- file buffering, 211
- file system, 208
- FilePath (type synonym), 99, 123
- filter, 114
- filterM, 199, 203
- find, 180, 181, 185
- findIndex, 180, 185
- findIndices, 180, 185
- fixity, 20
- fixity, 40, 56, 134
- fixity declaration, 50, 52, 56
- flip, 83, 110
- float, 12
- Float (datatype), 91, 94, 112
- floatDigits, 94, 95, 108
- Floating (class), 91, 93, 107
 - instance for Complex, 156
 - superclass of RealFloat, 108
- floating literal pattern, 35
- floatRadix, 94, 95, 108

- floatRange, 94, 95, 108
- floatToDigits, 158, 165
- floor, 94, 95, 107
- flushing a file buffer, 212
- fmap, 89, 109, 176
- foldl, 115
- foldl1, 115
- foldM, 199, 203
- foldr, 116
- foldr1, 116
- formal semantics, 3
- formatCalendarTime, 226, 229
- formatRealFloat, 164
- formfeed*, 9, 127
- fpat*, 33, 137
- fpats*, 33, 137
- Fractional (class), 21, 91, 93, 107
 - instance for Complex, 155
 - instance for Ratio, 151
 - superclass of Floating, 107
 - superclass of RealFrac, 107
- fromEnum, 88, 106
- fromInteger, 21, 92, 93, 106
- fromIntegral, 94, 96, 109
- fromJust, 191, 192
- fromMaybe, 191, 192
- fromRat, 157, 160
- fromRat', 160
- fromRational, 21, 92, 93, 107
- fst, 83, 113
- function, 83
- function binding, 58
- function type, 42, 43
- functional language, vii
- functor, 89
- Functor (class), 89, 109
 - instance for Array, 177
 - instance for IO, 111
 - instance for Maybe, 111
 - instance for [], 113
- funlhs*, 135
- gap*, 12, 128
- gcd, 93, 94, 108
- gcon*, 20, 137
- gconsym*, 20
- gd*, 27, 58, 135
- gdp*, 27, 136
- gdrhs*, 58, 135
- gendecl*, 40, 50, 56, 134
- generalization, 61
- generalization preorder, 44
- generator, 26
- genericDrop, 180, 187
- genericIndex, 180, 188
- genericLength, 180, 187
- genericReplicate, 180, 188
- genericSplitAt, 180, 188
- genericTake, 180, 187
- genRange, 235, 237
- get the contents of a file, 214
- getArgs, 223
- getChar, 98, 123
- getClockTime, 226, 228
- getContents, 98, 124
- getCPUTime, 233
- getCurrentDirectory, 220
- getDirectoryContents, 220
- getEnv, 223
- getLine, 98, 124
- getModificationTime, 220
- getPermissions, 220
- getProgName, 223
- getStdGen, 236, 240
- getStdRandom, 236, 240
- graphic*, 9, 127
- group, 180, 182, 186
- groupBy, 180, 183, 186
- GT, 83
- gtycon*, 42, 51, 134
- guard, 26, 28, 36
- guard, 199, 203
- Handle (datatype), 206
- HandlePosn (datatype), 206
- handles, 208
- Haskell, vii, 3
- Haskell kernel, 4
- hClose, 206, 210
- head, 115

- hexadecimal*, 11, 128
- hexit*, 9, 127
- hFileSize*, 206, 211
- hFlush*, 206, 212
- hGetBuffering*, 206, 212
- hGetChar*, 206, 214
- hGetContents*, 206, 209, 214
- hGetLine*, 206, 214
- hGetPosn*, 206, 213
- hiding*, 72
- Hindley–Milner type system, 4, 40, 60
- hIsClosed*, 207, 213
- hIsEOF*, 206, 211
- hIsOpen*, 207, 213
- hIsReadable*, 207, 213
- hIsSeekable*, 207, 213
- hIsWritable*, 207, 213
- hLookAhead*, 206, 212, 214
- hPrint*, 206, 215
- hPutChar*, 206, 215
- hPutStr*, 206, 215
- hPutStrLn*, 206
- hReady*, 206, 214
- hSeek*, 206, 213
- hSetBuffering*, 206, 212
- hSetPosn*, 206, 213
- hWaitForInput*, 206, 214

- I/O, 205
- I/O errors, 207
- id*, 83, 109
- idecl*, 40, 51, 134
- idecls*, 40, 51, 134
- identifier, 10
- if-then-else expression, *see* conditional expression
- imagPart*, 153, 154
- impdecl*, 71, 133
- impdecls*, 68, 133
- import*, 71, 133
- import declaration, 71
- impspec*, 71, 133
- index*, 169–171
- indices*, 173, 174, 177
- init*, 115
- inits*, 180, 182, 186
- inlining*, 145
- input/output*, 205
- input/output examples*, 215
- inRange*, 169–171
- insert*, 180, 182, 187
- insertBy*, 180, 183, 187
- inst*, 51, 135
- instance declaration, 51, 52, *see also* derived instance
 - importing and exporting, 74
 - with an empty where part, 51
- Int* (datatype), 91, 94, 111
- integer*, 12
- Integer* (datatype), 94, 112
- integer literal pattern*, 35
- integerLogBase*, 161
- Integral* (class), 91, 93, 106
- interact*, 98, 124
- intersect*, 180, 181, 185
- intersectBy*, 180, 183, 185
- intersperse*, 180, 182, 185
- intToDigit*, 193, 196
- IO* (datatype), 83, 111, 205
- IO* (module), 205, 216
- ioeGetErrorString*, 207, 208
- ioeGetFileName*, 207, 208
- ioeGetHandle*, 207, 208
- IOError* (datatype), 83, 123, 207
- ioError*, 100, 123
- IOMode* (datatype), 206, 210
- irrefutable pattern*, 27, 35, 36, 59
- isAlpha*, 193, 195
- isAlphaNum*, 193, 195
- isAlreadyExistsError*, 207
- isAlreadyInUseError*, 207, 210
- isAscii*, 193, 195
- isControl*, 193, 195
- isDigit*, 193, 195
- isDoesNotExistError*, 207, 210
- isEOF*, 206, 211
- isEOFError*, 207
- isFullError*, 207
- isHexDigit*, 193, 195
- isIllegalOperation*, 207

- isJust, 191, 192
- isLatin1, 193, 195
- isLower, 193, 195
- isNothing, 191, 192
- isOctDigit, 193, 195
- isPermissionError, 207, 208, 210
- isPrefixOf, 180, 186
- isPrint, 193, 195
- isSpace, 193, 195
- isSuffixOf, 180, 186
- isUpper, 193, 195
- isUserError, 207, 208
- iterate, 116
- Ix (class), 169–171
 - instance for Char, 171
 - instance for Integer, 171
 - instance for Int, 171
- Ix (module), 169, 171, 173, 176, 205, 225, 227
- Ix
 - derived instance, 170
- ixmap, 173, 176, 177
- join, 199, 203
- Just, 83
- kind, 41, 42, 46, 48, 52, 66
- kind inference, 42, 46, 48, 52, 66
- label, 29
- lambda abstraction, 21
- large, 9, 127
- last, 115
- layout, 13, 128, *see also* off-side rule
- lcm, 93, 94, 108
- Left, 83
- length, 115
- let expression, 27
 - in do expressions, 29
 - in list comprehensions, 25
- lex, 88, 121
- lexDigits, 158, 167
- lexeme, 9, 127
- lexical structure, 7
- lexLitChar, 193, 197
- lexⁱ, 18, 136
- libraries, 77
- liftM, 200, 203
- liftM2, 200, 203
- liftM3, 200, 203
- liftM4, 200, 203
- liftM5, 200, 203
- linear pattern, 21, 33, 58
- linearity, 21, 33, 58
- lines, 117
- list, 24, 42, 82
- List (module), 176, 179, 184
- list comprehension, 25, 82
- list type, 43
- listArray, 173, 174, 177
- listToMaybe, 191, 192
- literal, 9, 127
- literal pattern, 35
- literate comments, 131
- locale, 231
- Locale (module), 227, 231
- log, 93, 107
- logarithm, 93
- logBase, 93, 107
- lookahead, 214
- lookup, 118
- lpatⁱ, 33, 137
- LT, 83
- magnitude, 94
- magnitude, 153, 154
- Main (module), 67
- main, 67
- making directories, 220
- map, 114
- mapAccumL, 180, 182, 186
- mapAccumR, 180, 182, 186
- mapAndUnzipM, 199, 203
- mapM, 90, 109
- mapMaybe, 191, 192
- mapM_, 90, 109
- match, 196
- max, 86, 105, 140
- maxBound, 91, 106, 141
- maximal munch rule, 9, 13, 126
- maximum, 118

- maximumBy, 180, 183, 187
- Maybe (datatype), 83, 111
- Maybe (module), 184, 191, 192
- maybe, 83, 111
- maybeToList, 191, 192
- method, *see* class method
- min, 86, 105, 140
- minBound, 91, 106, 141
- minimum, 118
- minimumBy, 180, 183, 187
- mkPolar, 153, 154
- mkStdGen, 235, 237
- mod, 57, 92, 93, 105, 106
- modid, 11, 68, 128, 133
- module, 67
- module*, 68, 133
- monad, 29, 90, 97
- Monad (class), 29, 90, 109
 - instance for IO, 111
 - instance for Maybe, 111
 - instance for [], 113
 - superclass of MonadPlus, 202
- Monad (module), 199, 202
- MonadPlus (class), 199, 202
 - instance for Maybe, 202
 - instance for [], 202
- monomorphic type variable, 37, 62, 63
- monomorphism restriction, 63
- Month (datatype), 225, 227
- moving directories, 221
- moving files, 221
- mplus, 199, 202
- msum, 200, 203
- mzero, 199, 202

- $n+k$ pattern, 35, 59
- name
 - qualified, *see* qualified name
 - special, *see* special name
- namespaces, 5, 10
- ncomment*, 9, 127
- negate, 22, 92, 93, 106
- negation, 18, 22, 23
- newconstr*, 48, 135
- newline*, 9, 127

- newStdGen*, 236, 240
- newtype* declaration, 34, 35, 37, 48
- next, 235, 237
- nonnull, 167
- not, 81, 110
- notElem, 57, 114, 118
- Nothing, 83
- nub, 180, 181, 185
- nubBy, 180, 183, 185
- null, 115
- Num (class), 21, 54, 91, 93, 106
 - instance for Complex, 155
 - instance for Ratio, 151
 - superclass of Fractional, 107
 - superclass of Real, 106
- number, 91
 - literal syntax, 11
 - translation of literals, 20
- numerator, 149, 150
- Numeric (module), 119, 157, 160, 195
- numeric type, 92
- numericEnumFrom, 113
- numericEnumFromThen, 113
- numericEnumFromThenTo, 113
- numericEnumFromTo, 113

- octal*, 11, 128
- octit*, 9, 127
- odd, 93, 108
- off-side rule, 13, *see also* layout
- op*, 20, 56, 137
- opencom*, 9, 127
- openFile, 206, 210
- opening a file, 210
- operating system commands, 224
- operator, 10, 22
- operator application, 21
- ops*, 40, 56, 134
- or, 118
- Ord (class), 86, 91, 105
 - derived instance, 53, 140
 - instance for Array, 177
 - instance for Char, 110
 - instance for Ratio, 150
 - superclass of Real, 106

- ord, 193, 196
- Ordering (datatype), 83, 111
- otherwise, 81, 110
- overloaded functions, 40
- overloaded pattern, *see* pattern-matching
- overloading, 49
 - defaults, 53
- partition, 180, 182, 186
- patⁱ*, 33, 137
- pat*, 33, 137
- path, 220
- pattern, 28, 32
 - @, *see* as-pattern
 - _, *see* wildcard pattern
 - constructed, *see* constructed pattern
 - floating, *see* floating literal pattern
 - integer, *see* integer literal pattern
 - irrefutable, *see* irrefutable pattern
 - linear, *see* linear pattern
 - n+k*, *see* *n+k* pattern
 - refutable, *see* refutable pattern
- pattern binding, 58, 59
- pattern-matching, 32
 - overloaded constant, 37
- Permissions (datatype), 219
- phase, 153–155
- physical file, 208
- pi, 93, 107
- polar, 153, 154
- polling a handle for input, 214
- polymorphic recursion, 56
- polymorphism, 4
- pragmas, 145
- precedence, 45, *see also* fixity
- pred, 106
- Prelude
 - implicit import of, 77
- Prelude (module), 77, 104
- PreludeBuiltin (module), 105, 123
- PreludeIO (module), 105, 123
- PreludeList (module), 104, 105, 114
- PreludeText (module), 105, 119
- principal type, 44, 56
- print, 98, 123
- product, 118
- program*, 9, 127
- program arguments, 224
- program name, 224
- program structure, 3
- properFraction, 94, 95, 107
- putChar, 98, 123
- putStr, 98, 123
- putStrLn, 98, 123
- qcon*, 20, 137
- qconid*, 11, 128
- qconop*, 20, 137
- qconsym*, 11, 128
- qop*, 20, 22, 137
- qtycls*, 11, 128
- qtycon*, 11, 128
- qual*, 26, 136
- qualified name, 11, 72, 74
- qualifier, 26
- quantification, 43
- quot, 92, 93, 105, 106
- quotRem, 93, 106
- qvar*, 20, 137
- qvarid*, 11, 128
- qvarop*, 20, 137
- qvarsym*, 11, 128
- Random (class), 239
- Random (module), 235
- random, 236, 239
- random access files, 213
- RandomGen, 237
- randomIO, 236, 239
- randomR, 236, 239
- randomRIO, 236, 239
- randomRs, 236, 239
- randoms, 236, 239
- range, 169–171
- rangeSize, 169–171
- Ratio (datatype), 149
- Ratio (module), 105, 149, 150, 160
- Rational (type synonym), 149, 150
- rational numbers, 149
- Read (class), 87, 120
 - derived instance, 53, 142

- instance for Array, 178
- instance for Char, 122
- instance for Double, 122
- instance for Float, 122
- instance for Integer, 122
- instance for Int, 121
- instance for Ratio, 151
- instance for [a], 122
- read, 87, 88, 120
- readable, 219
- readDec, 157, 162
- readEsc, 196
- readFile, 99, 124, 210
- readFloat, 158, 167
- readHex, 157, 162
- reading a directory, 221
- reading from a file, 214
- readInt, 157, 162
- readIO, 98, 124
- readList, 87, 120, 142
- readLitChar, 193, 196
- readLn, 98, 124
- ReadMode, 210
- readOct, 157, 162
- readParen, 120
- ReadS (type synonym), 87, 119
- reads, 87, 88, 120
- readSigned, 157, 162
- readsPrec, 87, 120, 142
- ReadWriteMode, 210
- Real (class), 91, 93, 106
 - instance for Ratio, 151
 - superclass of Integral, 106
 - superclass of RealFrac, 107
- RealFloat (class), 94, 95, 108
- RealFrac (class), 94, 107
 - instance for Ratio, 151
 - superclass of RealFloat, 108
- realPart, 153, 154
- realToFrac, 94, 96, 109
- recip, 93, 107
- recursive datatype, 48
- refutable pattern, 35
- RelativeSeek, 213
- rem, 57, 92, 93, 105, 106
- removeDirectory, 219, 220
- removeFile, 219, 220
- removing directories, 220
- removing files, 220
- renameDirectory, 219, 220
- renameFile, 219, 220
- renaming directories, 221
- renaming files, 221
- repeat, 116
- replicate, 116
- reservedid*, 10, 128
- reservedop*, 10, 128
- return, 90, 109
- reverse, 118
- rexpⁱ*, 18, 136
- rhs*, 58, 135
- Right, 83
- round, 94, 95, 107
- roundTo, 164
- rpatⁱ*, 33, 137
- scaleFloat, 94, 108
- scaleRat, 161
- scanl, 116
- scanl1, 116
- scanr, 116
- scanr1, 116
- scontext*, 134
- searchable, 219
- section, 10, 22, *see also* operator application
- SeekFromEnd, 213
- seeking a file, 213
- SeekMode (datatype), 206, 213
- semantics
 - formal, *see* formal semantics
- semi-closed handles, 209
- separate compilation, 78
- seq, 84, 105, 110
- sequence, 90, 109
- sequence_, 90, 109
- setCurrentDirectory, 220
- setPermissions, 220
- setStdGen, 236, 240
- setting the directory, 221
- Show (class), 87, 120

- derived instance, 53, 142
- instance for Array, 178
- instance for Char, 122
- instance for Double, 122
- instance for Float, 122
- instance for HandlePosn, 206
- instance for Integer, 121
- instance for Int, 121
- instance for Ratio, 151
- instance for [a], 122
- superclass of Num, 106
- show, 87, 88, 120
- show2, 230
- show3, 230
- showChar, 120
- showEFloat, 158, 162
- showFFloat, 158, 162
- showFloat, 158, 162
- showGFloat, 158, 162
- showHex, 157, 161
- showInt, 157, 161
- showIntAtBase, 157, 162
- showList, 87, 120, 142
- showLitChar, 193, 197
- showOct, 157, 161
- showParen, 120
- ShowS (type synonym), 87, 119
- shows, 87, 88, 120
- showSigned, 157, 161
- showsPrec, 87, 120, 142
- showString, 120
- sign, 94
- signature, *see* type signature
- signdekl*, 55
- significand, 94, 95, 108
- signum, 93, 94, 106
- simple pattern binding, 59, 64
- simpleclass*, 44, 134
- simpletype*, 45, 47, 48, 135
- sin, 93, 107
- sine, 95
- sinh, 93, 107
- size of file, 211
- small*, 9, 127
- snd, 83, 113
- sort, 180, 182, 187
- sortBy, 180, 183, 187
- space*, 9, 127
- span, 117
- special*, 9, 127
- split, 235, 237
- splitAt, 117
- sqrt, 93, 107
- standard handles, 209
- standard prelude, 77, *see also* Prelude
- stderr, 206, 209
- StdGen (datatype), 235, 237
- stdin, 206, 209
- stdout, 206, 209
- stmt*, 29, 136
- stmts*, 29, 136
- strictness flag, 47
- strictness flags, 84
- string, 82
 - literal syntax, 12
 - transparent, *see* transparent string
- string*, 12, 128
- String (type synonym), 82, 111
- subtract, 108
- succ, 106
- sum, 118
- superclass, 50, 51
- symbol*, 9, 127, 128
- synonym, *see* type synonym
- syntax, 125
- System (module), 223
- system, 223
- tab*, 9, 127
- tail, 115
- tails, 180, 182, 186
- take, 117
- takeWhile, 117
- tan, 93, 107
- tangent, 95
- tanh, 93, 107
- tdDay, 226, 228
- tdHour, 226, 228
- tdMin, 226, 228
- tdMonth, 226, 228

- tdPicosec, 226, 228
- tdYear, 226, 228
- terminating a program, 224
- the file system, 219
- time, 225
- Time (module), 219, 225, 227
- time of day, 225
- time12Fmt, 231
- TimeDiff (datatype), 226, 228
- timeFmt, 231
- TimeLocale (datatype), 231
- to12, 229
- toCalendarTime, 226, 228
- toClockTime, 226, 228
- toEnum, 88, 106
- toInteger, 106
- toLower, 193, 196
- topdecl* (class), 50
- topdecl* (data), 45
- topdecl* (default), 53
- topdecl* (instance), 51
- topdecl* (newtype), 48
- topdecl* (type), 47
- topdecl*, 40, 133
- topdecls*, 40, 68, 133
- toRational, 93, 95, 106
- toUpper, 193, 196
- toUTCTime, 226, 228
- transpose, 180, 182, 186
- trigonometric function, 95
- trivial type, 25, 42, 83
- True, 81
- truncate, 94, 95, 107
- try, 207, 216
- tuple, 24, 42, 82
- tuple type, 43
- tycls*, 11, 44, 128
- tycon*, 11, 128
- type, 4, 41, 44
 - ambiguous, *see* ambiguous type
 - constructed, *see* constructed type
 - function, *see* function type
 - list, *see* list type
 - monomorphic, *see* monomorphic type
 - numeric, *see* numeric type
 - principal, *see* principal type
 - trivial, *see* trivial type
 - tuple, *see* tuple type
- type*, 42, 134
- type class, 4, 40, *see* class
- type constructor, 45
- type environment, 44
- type expression, 42
- type renaming, *see* newtype declaration
- type signature, 44, 52, 55
 - for an expression, *see* expression type-signature
- type synonym, 47, 51, 70, *see also* datatype
 - recursive, 48
- tyvar*, 11, 44, 128
- uncurry, 83, 113
- undefined, 19, 114
- unfoldr, 180, 187
- Unicode character set, 8, 12
- UnicodePrims (module), 105, 195
- uniDigit*, 9, 127
- uniLarge*, 9, 127
- union, 180, 181, 185
- unionBy, 180, 183, 185
- uniSmall*, 9, 127
- uniSymbol*, 9, 127
- unit datatype, *see* trivial type
- unit expression, 25
- uniWhite*, 9, 127
- unless, 199, 203
- unlines, 118
- until, 83, 113
- unwords, 118
- unzip, 119
- unzip3, 119
- unzip4, 180, 184, 188
- unzip5, 180, 189
- unzip6, 180, 189
- unzip7, 180, 189
- userError, 100, 123
- valdefs*, 51
- value, 4
- var*, 20, 137
- varid*, 10, 11, 128

varop, 20, 137
vars, 40, 55, 134
varsym, 10, 128
vertab, 9, 127

when, 199, 203
whitechar, 9, 127
whitespace, 9, 127
whitestuff, 9, 127
wildcard pattern (`_`), 33
words, 118
writable, 219
writeFile, 99, 124, 210
WriteMode, 210

zip, 83, 119
zip3, 119
zip4, 180, 184, 188
zip5, 180, 188
zip6, 180, 188
zip7, 180, 188
zipWith, 119
zipWith3, 119
zipWith4, 180, 184, 188
zipWith5, 180, 188
zipWith6, 180, 188
zipWith7, 180, 188
zipWithM, 199, 203
zipWithM_, 199, 203