Index

* +int → int → int, 27
  δ-rule, see delta-rules
  λ-abstractions, see lambda abstractions
  +, see addition

abstract domain
  optimal, 100
  using complete lattices, 42, 63–65, 70, 89
abstract interpretation, 15, 20, 50, 158
  \(\Lambda_T\), 65–67
application, 66
arithmetic functions, 83, 181
  bool, 65, 78, 105
boolean functions, 83, 181
comparison functions, 83, 181
  const\(\tau\) → (list \(\tau\)) → (list \(\tau\)), 83, 94, 97, 99, 101, 183
  constants, 73, 77, 82, 94, 97, 99, 101
  correct, 12, 13, 16, 64, 74, 88
empty\((tree \(\tau\))\), 83
  example functions, see under each function name
false\(\text{bool}\), 83
  fix\(\sigma\ → \sigma\), 66
  head\((list \(\tau\)) → \(\tau\), 83, 94, 97, 99, 101, 182
  if\(\text{bool} → \sigma → \sigma\), 82, 83
  ins\(l(\sigma_1, ... \sigma_k)\), 65, 66
  int, 65, 78, 105
  integers, 83
  iss\(l\sigma_1, ..., \sigma_k → \text{bool}\), 188
  lambda abstraction, 66
  lcase\((list \(\tau\)) → \sigma → (\tau → (list \(\tau\)) → \sigma), 83, 86, 94, 97, 99, 101
  lift\(\sigma\), 66

(list \(\tau\)), 65, 79, 80, 92, 97, 99, 102, 105
  nil\((list \(\tau\))\), 83, 94, 97, 99, 101
  node\((tree \(\tau\)) → \(\tau → (tree \(\tau\)) → (tree \(\tau\))\), 83
  outs\(t\sigma_1, ..., \sigma_k → \sigma\), 83, 188
  right\((tree \(\tau\)) → (tree \(\tau\))\), 83, 185
  rule of signs, 1, 9
  smash sum, 51, 65
  example, 84
tail\((list \(\tau\)) → (list \(\tau\))\), 83, 85, 94, 97, 99, 101
  take\(l(\sigma_1, ..., \sigma_k) → \sigma\), 53, 83, 186
  takes\(l\sigma_1, ..., \sigma_k → \sigma\), 53, 83, 187
  tcase\((tree \(\tau\)) → \sigma → ((tree \(\tau\)) → \(\tau → (tree \(\tau\)) → \sigma\)), 83, 186
  (tree \(\tau\)), 65, 80, 102, 105
  true\(\text{bool}\), 83
tuple\((\sigma_1, ..., \sigma_k)\), 66
tuples\((\sigma_1, ..., \sigma_k)\), 66
types, 66
  value\((tree \(\tau\)) → \tau\), 83, 184
  variables, 66

abstraction map, 13, 67
  adjointness, 71
  alternative definitions, 70
  bottom-reflexivity of, 71, 81, 93, 98
  definition of, 68, 69, 79, 92, 102
  properties of, 71
  reduced product, 98
  relationship with concretisation map, 15, 70
  semi-homomorphism, 72, 73, 82
  strictness of, 67, 70, 81, 93, 98
ADD, 131, 196
add, 131
addition
  compilation of, 131–133
  reduction, 27, 119, 122
adjoined functions, 69, 71
algebraic, 44
AND, 196
and \textit{bool}→\textit{bool}
abstract interpretation, \textit{see} abstract interpretation, boolean functions
delta-rules, 34, 35
standard interpretation, \textit{see} standard interpretation, boolean functions
strictness of, 35, 58
append, 6
abstract interpretation, 91
abstract interpretation using $A_{\text{list}}^h$, 94
evaluation transformers for, 114
translation into $\Lambda_T$, 37
application, 21
abstract interpretation, 66
annotating with evaluator transformers, 108
compilation of, 127, 129, 137
evaluation of, 124
general interpretation, 53
left-associativity, 21
standard interpretation, 56
stored as Vaps, 123
using textual juxtaposition, 21
arithmetic functions
abstract interpretation, 181
delta-rule, 34
standard interpretation, 60
\textit{arithmetic\_op}, 198
\textit{assoc}, 209

backwards analysis, 154
\textit{before}, 6
abstract interpretation, 91
abstract interpretation using $A_{\text{list}}^h$, 94
evaluation transformers for, 114
translation into $\Lambda_T$, 37
big union, 49
binding, 23, 33
boolean functions
abstract interpretation, 181
delta-rule, 34
standard interpretation, 60
\textit{boolean\_op}, 198
bottom, 11, 42
bottom-reflecting, 43, 67, 71, 81, 93, 98, 174
bound variable, 23, 33
\textit{C}, \textit{see} compilation scheme, \textit{C}
c\textit{r}, \textit{see} Index of Symbols
\textit{CALL}, 133, 146, 197
call, 206
call-by-name, 3
chain, 40
Church-Rosser property, 24, 29, 36, 161, 208
failure of, 29, 208
Church-Rosser Theorem, 25, 36, 62, 157
closed term, 23, 61, 136, 166-168
closure, 9, 117, 119, 159
Scott, \textit{see} Scott-closed
code
and context-free evaluation transformers, 136
and context-sensitive evaluation transformers, 136
entry point, 145-147
optimisation, 132, 133, 139
parallel, 141-143, 159
sequential, 137-140, 159
combinator, 23, 118
reduction, 4, 118, 136
compact, 44
comparison functions
abstract interpretation, 181
delta-rule, 34
standard interpretation, 60
\textit{comparison\_op}, 198

compilation scheme
\textit{C}, 129, 139, 193
\textit{E}, 133, 137, 192
\textit{F}, 124, 191
\textit{P}, 141
\textit{R}, 124, 133, 137, 140, 191
strictness and \textit{E}, 133, 134, 139
complete lattice, 42
complete partial order, 42
consistently complete, 44
flat, 43, 55, 57
Computational Adequacy Theorem, 20, 61, 78, 79, 106, 108, 158
concretisation, 12, 64
concretisation map, 12, 13, 67
adjointness, 71
alternative definitions, 71
definition of, 69
properties of, 71
relationship with abstraction map, 15, 70
strictness of, 71, 81, 93, 98, 108
well-definedness of, 70
conditional, see $\text{if}_{\text{bool}} \rightarrow \sigma \rightarrow \sigma \rightarrow \sigma$
CONS, 128, 196
Cons, 128
\text{cons}_{\tau \to (\text{list } \tau) \to (\text{list } \tau)}, 32
abstract interpretation, 83, 94, 183
abstract interpretation using $A^{\text{hs}}_{\text{list } \tau}$, 99
abstract interpretation using $A^{\text{rp}}_{\text{list } \tau}$, 97
abstract interpretation using lifting, 101
 compilation of, 128, 140
evaluation transformers for, 114
standard interpretation, 60
cons, 58, 183
cons, 128, 199
constant, 20, 27, 28, 32, 34, 73
abstract interpretation, see abstract interpretation, constants
general interpretation, 53
induced interpretation, 73
standard interpretation, see standard interpretation, constants
constant safety condition, 73, 74, 77, 82, 93, 102, 158, 178
Context-Free Evaluation Transformer Theorem, 112
Context-Sensitive Evaluation Transformer Theorem, 108, 137
continuous function, 46
Correctness Theorem for Abstract Interpretation, 74, 88, 93, 95, 98, 159
cpo, see complete partial order
curry, 73
delta-rules, 20, 27, 28, 34, 118
denotational semantics, 15, 19, 39, 50, 158
relation to operational semantics, 61
directed set, 43
DIV, 196
domain, 11, 44
coalesced sum, 45
flat, see complete partial order, flat
lifted, 46
powerdomain, 48
product, 44, 65
smash product, 45
smash sum, 45, 65
sum, 45
not needed, 46
domain theory, 11, 39
doubles, 6
abstract interpretation, 91
abstract interpretation using $A^{\text{hs}}_{\text{list } \tau}$, 94
evaluation transformers for, 114
translation into $\Lambda_T$, 37
$E$, see compilation scheme, $E$
e.lookup, 210
\text{empty}_{\text{tree } \tau}, 32
abstract interpretation, 83
delta-rule, 34
standard interpretation, 60
\text{empty}, 58
ENTER, 127, 129, 146, 147, 197
enter, 205
environment, 53
modified, 53
EQ, 196
eq, 198
EVAL, 123, 124, 128, 131, 133, 146, 197
eval, 124, 206
evaluation transformer, 1, 7, 8, 78, 108, 155, 159
collection-free, 112, 118, 136, 144, 146, 153, 159
collection-sensitive, 112, 118, 136, 147, 153, 159
evaluator, 7, 16, 159
   as label on graph node, 145
definition of, 107, 108
parallel implementation, 143
sequential implementation, 140

F, see compilation scheme, F
falsebool, 27
   standard interpretation, 55, 60
false, 55, 57
false
   abstract interpretation, 83
find_destination, 201
finite element, 44
fix (σ→σ→), 27, 29
   abstract interpretation, 66
general interpretation, 54
standard interpretation, 56
fixed point, 29, 43, 54
   least, 43, 54, 89
Fixed Point Theorem, 43, 54
Ff, see reduction strategy, Ff
flat, 43
forwards analysis, 154
free variable, 23, 35
function
   bottom-reflecting, 43
   composition, 46
   continuous, 43
   lambda abstraction, see lambda abstraction
   monotonic, 41
   strict, 43, 133
functions
   adjoined, 69, 71
g_lookup, 209
GE, 196
general interpretation, 50–55

application, 52, 53
bool, 51
constant, 53
fix (σ→σ→), 52, 54
ins (σ1→,...,σk), 52, 54
int, 51
lambda abstraction, 52, 53
let-expression, 54
lift, 51
lift, 52
product, 51
smash product, 51
smash sum, 51
tuple (σ1→,...,σk), 52, 54
tuples (σ1→,...,σk), 52, 54
type, 51, 52
variable, 52, 53
graph, 119
   body of function, 121
   building of, 121, 129
graph node
   Cons, 128, 143–146
   Int, 121, 131
   Vap, 119, 129, 144, 145
   Var, 121
graph reduction, 3, 117, 119, 122
ground type, 61
GT, 196

HEAD, 196
head, 3
head (list τ→τ), 32
   abstract interpretation, 83, 94, 182
   abstract interpretation using A_{list τ}, 99
   abstract interpretation using A^{lp}_{list τ}, 97
   abstract interpretation using lifting, 101
delta-rule, 33, 34
evaluation transformers for, 114
standard interpretation, 60
head, 59, 183
head, 199
head normal form, 4, 25, 27, 36, 118, 122, 129, 205–207
  non-uniqueness, 25
head normalising, 165
head redex, 25
Head-Normalisation Theorem, 29, 37, 105, 117, 157
head-reduction, 36
HNF, see head normal form
Hoare powerdomain, 49

if\textsubscript{bool→σ→σ}, 27
  abstract interpretation, 82, 83
  compilation of, 134, 140, 141
delta-rule, 28, 34
  standard interpretation, 60
Index of Symbols, 224
index\_stack, 209
induced interpretations, 73
infinite computation, 61, 106, 158
  initiating an, 106
infinite list, 4, 80
infinite reduction sequence, 106
information content, 39, 69
  and monotonic functions, 41
  bottom, 42, 61
  infinite computation, 61
  least upper bound, 41, 68
\textit{in}\textit{HNF}, 25, 33, 34, 36
\textit{ins}_i(σ_1,...,σ_k), 33
  abstract interpretation, 65, 66
general interpretation, 54
  standard interpretation, 55, 56
\textit{ins}_i, 55
instruction set, 195
  definition of instructions, 196
interpretation, 10, 15, 20, 50, 158
  abstract, see abstract interpretation
general, see general interpretation
standard, see standard interpretation
type
  abstract, see abstract interpretation, type
general, see general interpretation, type

standard, see standard interpretation, type
inverse comment convention, 1, 118
is\textit{bool}, 208
is\textit{Cons}, 209
is\textit{Empty}, 209
is\textit{Fun}, 208
is\textit{Int}, 208
is\textit{Nil}, 209
is\textit{Node}, 209
\textit{iss}_i:σ_1⊗...⊗σ_k→\textit{bool}, 32
  abstract interpretation, 188
delta-rule, 34
  standard interpretation, 60
\textit{iss}_i, 47, 188
is\textit{Vap}, 208
is\textit{Wvap}, 208
J\textit{FALSE}, 197
\textit{jfalse}, 201
JMP, 197
\textit{jmp}, 201
join, 6
  abstract interpretation, 91
evaluation transformers for, 114
  translation into Λ\textit{T}, 37
LABEL, 197
lambda abstraction, 21
  abstract interpretation, 66
general interpretation, 53
  standard interpretation, 56
lambda calculus
  untyped, 27
lattice, 12, 42
  for abstract domain, 63–65, 70, 89
lazy evaluation, 1, 4, 119
  and graph reduction, 16, 119
code for, 17, 123
correct implementation of, 136
  restrictiveness of, 1
lazy functional languages, 15
  application using textual juxtaposition, 2
type inference, 7
lazy semantics
not preserved, 155
LCASE, 197
lcase\((\text{list } \tau) \rightarrow \sigma \rightarrow \tau \rightarrow (\text{list } \tau) \rightarrow \sigma \rightarrow \sigma\), 32
abstract interpretation, 83, 86, 94
abstract interpretation using \(A_{\text{hs}}^{\text{hts}}\), 99
abstract interpretation using \(A_{\text{rp}}^{\text{hts}}\), 97
abstract interpretation using lifting, 101
compilation of, 140
delta-rule, 34
standard interpretation, 60
lcase, 59, 86
lcase, 202
LE, 196
LEFT, 197
left\((\text{tree } \tau) \rightarrow (\text{tree } \tau)\), 32
abstract interpretation, 83, 185
delta-rule, 34
evaluation transformers for, 114
standard interpretation, 60
left, 59, 185
left, 201
left-closed set, 49, 170
leftmost outermost reduction, 3
length, 6
abstract interpretation, 91
abstract interpretation using \(A_{\text{hs}}^{\text{hts}}\), 99
evaluation transformers for, 114
translation into \(\Lambda_T\), 37
left-expression, 33
abstract interpretation, 66, 67
delta-rule, 33, 34
general interpretation, 54
standard interpretation, 58
lift\(\sigma\), 33
abstract interpretation, 66
general interpretation, 54
standard interpretation, 56
lift, 46
and abstract domains for lists, 101
list
infinite, 113
and abstract domains for lists, 101

partial, 80, 113
\((\text{list } \tau)\)
abstract interpretation, see abstract interpretation, \((\text{list } \tau)\)
standard interpretation, see standard interpretation, \((\text{list } \tau)\)
LOAD, 146, 197
load, 205
lower bound, 42
greatest, 42
LT, 196

map, 6
abstract interpretation, 91
abstract interpretation using \(A_{\text{hs}}^{\text{hts}}\), 94
evaluation transformers for, 114
translation into \(\Lambda_T\), 37
\(\text{mk}_\text{sum}\), 47, 55
monotonic, 41
MUL, 196

newlabel, 128, 209
next_to_top, 209
\(\text{nil}_{\text{list } \tau}\), 32, 34
abstract interpretation, 83, 94
abstract interpretation using \(A_{\text{hs}}^{\text{hts}}\), 99
abstract interpretation using \(A_{\text{rp}}^{\text{hts}}\), 97
abstract interpretation using lifting, 101
standard interpretation, 60
\(\text{nil}\), 58
NODE, 197
node\((\text{tree } \tau) \rightarrow \tau \rightarrow (\text{tree } \tau) \rightarrow (\text{tree } \tau)\), 32
abstract interpretation, 83
evaluation transformers for, 114
standard interpretation, 60
node, 58
node, 200
normal form, 25, 27
one-level head-strict, 77, 92
operational semantics, 15, 19
relation to denotational semantics, 61
optimal abstract domain, 100
optimisation of code, 132, 133, 139
OR, 196

\[ \text{or}_{\text{bool}-\text{bool}-\text{bool}} \]
abstract interpretation, see abstract interpretation, boolean functions
delta-rules, 35
standard interpretation, see standard interpretation, boolean functions
strictness of, 35, 58

\[ \text{outs}_{\sigma_1 \ldots \sigma_k \rightarrow \sigma_i}, 32 \]
abstract interpretation, 83, 188
delta-rule, 34
standard interpretation, 60
outs_i, 47, 188

\[ \mathcal{P}, \text{ see compilation scheme, } \mathcal{P} \]
parallel code, 141-143
partial application, 129, 147
partial list, 80
partial order, 11, 39
diagram, 40
flat, see complete partial order, flat
pattern matching, 4
overlapping patterns, 208
polymorphic, 5
polymorphically invariant, 153
powerdomain, 48
algebraic properties, 49
ordering, 48
PRINT, 197
print, 207
print_state, 210
program, 61, 106, 129, 136
program transformation, 139, 141
projection, 154
projection analysis, 154
Prolog, 152
PUSH, 131, 197
push, 203
PUSHBOOL, 197
pushbool, 203
PUSHEMPTY, 197
pushempty, 200
PUSHFUN, 127, 145, 197
pushfun, 203
PUSHINT, 197
pushint, 203
PUSHNIL, 196
pushnil, 199

\[ \mathcal{R}, \text{ see compilation scheme, } \mathcal{R} \]
recursion, 27
recursion and \( \text{fix}_{(\sigma \rightarrow \sigma) \rightarrow \sigma} \), 27, 38
recursive
domain isomorphism, 57
type, see type, recursive
redex, 2, 24
leftmost, 3, 37
outermost, 3
to the left of, 36, 163
reduced product, 96
reduces, 2

reduction
\( \beta \), 20, 24
\( \beta\delta \), 29, 136
algorithm, 121
combinator, 4, 118, 136
evaluation transformer model, 1, 7, 8, 136, 137, 155
graph, see graph reduction
one-step \( \beta \), 23
one-step \( \beta\delta \), 29
one-step \( \delta \), 27

reduction sequence
infinite, 106
standard, 163
strongly equivalent, 163
strongly standard, 162-164

reduction strategy, 26, 62, 139
conservative, 106
\( F_t \), 37, 61, 62, 161, 164, 165
head normalising, 26, 36, 161
head-, 26
leftmost, 33
leftmost outermost, 3
one-step, 26
safe, 61, 106, 115, 158
sequential, 61
speculative, 107
related, 107
represent, 64, 68
residual, 162
RETURN, 128, 131, 197
return, 128, 207
reverse, 6
abstract interpretation, 91
abstract interpretation using $A_{(\text{list } \tau)}^{A_{\tau}}$, 94
evaluation transformers for, 114
translation into $\Lambda_T$, 37
rewrites, 2
rewriting, 2
RIGHT, 197
right$_{(\text{tree } \tau)\rightarrow(\text{tree } \tau)}$, 32
abstract interpretation, 83, 185
delta-rule, 34
evaluation transformers for, 114
standard interpretation, 60
right, 59
right, 201
rule of signs, 9, 78
run, 196
safe, 61, 105, 106
Scott-closed, 49, 170
semi-homomorphism, 14, 72
sequential code, 137–140
sharing, 119
size, 6
abstract interpretation, 91
evaluation transformers for, 114
translation into $\Lambda_T$, 37
smash sum
abstract interpretation, see abstract interpretation, smash sum
domain, see domain, smash sum
standard interpretation, see standard interpretation, smash sum
SPAWN, 141, 146
Spineless G-machine, 123–136
SQUEEZE, 127, 136, 197
squeeze, 127, 204
stack frame, 124, 136
stack invariants, 125, 131, 146, 204
stack_pop, 209
stack_push, 209
standard interpretation, 10, 15, 20, 50, 158
$\Lambda_T$, 55–58
application, 56
arithmetic functions, 60
bool, 55, 57
boolean functions, 60
comparison functions, 60
cons$_{(\text{list } \tau)\rightarrow(\text{list } \tau)}$, 60
constants, 58
empty$_{(\text{tree } \tau)}$, 60
false$_{\text{bool}}$, 55, 60
fix$_{(\sigma\rightarrow\sigma)\rightarrow\sigma}$, 56
head$_{(\text{list } \tau)\rightarrow\tau}$, 60
if$_{\text{bool}\rightarrow\sigma\rightarrow\sigma}$, 60
ins$_{1(\sigma_1,\ldots,\sigma_k)}$, 55, 56
int, 55, 57
iss$_{1(\sigma_1,\ldots,\sigma_k)\rightarrow\text{bool}}$, 60
lambda abstraction, 56
lcase$_{(\text{list } \tau)\rightarrow\sigma\rightarrow(\text{list } \tau)\rightarrow\sigma\rightarrow\sigma}$, 60
left$_{(\text{tree } \tau)\rightarrow(\text{tree } \tau)}$, 60
let-expression, 58, 60
lift$_{\sigma}$, 56
(list $\tau$), 55, 57
nil$_{(\text{list } \tau)}$, 60
node$_{(\text{tree } \tau)\rightarrow\tau\rightarrow(\text{tree } \tau)\rightarrow(\text{tree } \tau)}$, 60
outs$_{1(\sigma_1,\ldots,\sigma_k)\rightarrow\sigma_1}$, 60
right$_{(\text{tree } \tau)\rightarrow(\text{tree } \tau)}$, 60
smash sum, 55
tail$_{(\text{list } \tau)\rightarrow(\text{list } \tau)}$, 60
take$_{1(\sigma_1,\ldots,\sigma_k)\rightarrow\sigma_1}$, 53, 58
takes$_{1(\sigma_1,\ldots,\sigma_k)\rightarrow\sigma_1}$, 53, 58
tcase$_{(\text{tree } \tau)\rightarrow\sigma\rightarrow((\text{tree } \tau)\rightarrow\tau\rightarrow(\text{tree } \tau)\rightarrow\sigma)\rightarrow\sigma}$, 60
(tree $\tau$), 55, 57
true$_{\text{bool}}$, 55, 60
tuple$_{(\sigma_1,\ldots,\sigma_k)}$, 56
tuples$_{(\sigma_1,\ldots,\sigma_k)}$, 56
type, 56
value$_{(\text{tree } \tau)\rightarrow\tau}$, 60
variable, 56
standard reduction sequence, 163
Standardisation Theorem, 163, 164
step, 196
STOP, 197
STORE, 129, 146, 197
store, 129, 203
strict, 43, 98
strict function
  compilation of, 133
strictness analysis, 151–153
strongly equivalent, 163
strongly normalising, 27
strongly standard reduction sequence, 163
SUB, 196
substitution, 23, 35
sumlist, 6
  abstract interpretation, 91
determination of, 88–90
  abstract interpretation using $A^{ks}_{(\text{list } \tau)}$, 94
evaluation transformers for, 114
translation into $\Lambda T$, 37, 88
sumtree, 6
  abstract interpretation, 91
evaluation transformers for, 114
translation into $\Lambda T$, 37
tag, 121
TAIL, 196
tail, 3
tail$_{(\text{list } \tau)\rightarrow(\text{list } \tau)}$, 32
  abstract interpretation, 83, 85, 94
  abstract interpretation using $A^{hs}_{(\text{list } \tau)}$, 99
  abstract interpretation using $A^{hp}_{(\text{list } \tau)}$, 97
  abstract interpretation using lifting, 101
delta-rule, 34
evaluation transformers for, 114
standard interpretation, 60
tail, 59, 85
tail, 199
tail-recursive, 136
tail-strict, 95
take$_{\sigma_1\times\ldots\times\sigma_k\rightarrow\sigma}$, 32
  abstract interpretation, 53, 53, 186
delta-rule, 34
standard interpretation, 53, 58
takes$_{\sigma_1\circ\ldots\circ\sigma_k\rightarrow\sigma}$, 32
  abstract interpretation, 53, 83, 187
delta-rule, 34
standard interpretation, 53, 58
TCASE, 197
tcase$_{(\text{tree } \tau)\rightarrow\sigma\rightarrow(\text{tree } \tau)\rightarrow(\text{tree } \tau)\rightarrow\sigma}$, 32
  abstract interpretation, 83, 186
  compilation of, 140
delta-rule, 34
standard interpretation, 60
tcase, 59
tcase, 202
term graph rewriting, 122
Tim, 139
to the left of, 36, 163
top, 42
top, 209
total order, 39
trace, 210
(tree $\tau$)
  abstract interpretation, see abstract interpretation, (tree $\tau$)
  standard interpretation, see standard interpretation, (tree $\tau$)
true$_{\text{bool}}$, 27
  standard interpretation, 55, 60
true, 55, 57
tuple$_{(\sigma_1,\ldots,\sigma_k)}$, 30
  abstract interpretation, 66
general interpretation, 54
  standard interpretation, 56
tuples$_{(\epsilon_1,\ldots,\epsilon_k)}$, 33
  abstract interpretation, 66
delta-rule, 33, 34
general interpretation, 54
  standard interpretation, 56
type
  abstract interpretation, see abstract interpretation, types
checking, 20
ground, 61
inference, 155
lifted, 30
Y, 29

monomorphic, 5
polymorphic, 5, 153

product, 30
recursive, 8, 20, 33, 51, 78, 158
smash product, 30
smash sum, 30
standard interpretation, see standard interpretation, type
valid, 61, 129, 136
type variable, 5, 22
typed lambda calculus
  monomorphic, 22
  polymorphic, 22
  second-order, 22
  simple, 22

uncurry, 73
UPDATE, 124, 197
update, 207
update analysis, 151
UPDATEEVT, 145, 147
upper bound, 41
  least, 41, 64
valid type, 61
VALUE, 197

\text{value}_{(\text{tree } r) \to r}
  abstract interpretation, 83, 184
delta-rule, 34
evaluation transformers for, 114
standard interpretation, 60

value, 59, 184
value, 201
variable
  abstract interpretation, see abstract interpretation, variables
  bound, 23
  compilation of, 130, 144
  free, 23
general interpretation, see general interpretation, variable
standard interpretation, see standard interpretation, variable
variables
  standard interpretation, 56
version, 137

compile-time choice, 143, 159
run-time choice, 159

\omega\text{-algebraic, 44}
weak head normal form, 4
WSTORE, 197
wstore, 204

Y, 29