

Index

- () (0-tuple), 19
- [] (lists), 37, 41, 45, 50
- { }
 - in atomic expression, 35
- { } (finite map), 13, 35
- (* *) (comment brackets), 147
- , (comma), 35, 58
- ... (wildcard pattern row), 54
- _ (underbar)
 - in identifier, 63
 - in type variable, 45
 - wildcard pattern, 18
- |, 18
- = (see equality)
- =>, 19, 38
- >, 26, 41
- ~, 54
- . (period)
 - in real constants, 2
 - in long identifiers, 57
- ", 3
- ! (dereferencing), 41, 45
- +, 54
- , 54
- : (see type constraint and signature ascription)
- ::, 37, 45, 148
- := (assignment), 16, 41, 45
- <, 54
- >, 54
- <=, 54
- >=, 54
- >:, 84
- <:, 85
- *, 54, 61
- { } (empty map), 2
- + (see modification)
- ⊕, 3, 36, 78, 95
- ∪ (union, sometimes disjoint), 13
- × (cartesian product), 13
- ∇ (inference tree), 3, 98, 99, 118
- Λ (in type function), 48
- ∀ (in type scheme), 36
 - see also quantification
- α (type variable), 36
- η-conversion, 46
- μ (substitution), 53, 100
- ρ (record type), 35, 54
- τ (type), 36
- σ (see type scheme)
- ∀α^(k).τ (see type scheme)
- (function type), 36
- ↓ (restriction), 30, 33, 135
- θ (type function), 47, 48, 62, 66, 140
- (t, CE), 48, 51, 75, 112
- (θ, CE) (type structure), 6, 47–49, 89, 97, 138, 145
- Λα^(k).τ (type function), 47, 48, 62, 66, 140
- Σ (signature), 59–63
- (-)- (binding operator), 61
- (N)S (see also signature), 59
- Φ (functor signature), 76, 78, 80–83, 86, 92, 95, 141–143
- (N)(S, (N')S') (see also functor signature), 76
- φ_{Ty} (type realisation), 70, 111
- φ_{Str} (structure realisation), 70
- φ (see realisation)
- φB, 99
- φN, 98
- φT, 98
- ≥ (see instance)
- ⋃ (see generalisation and enrichment)
- ⊢ (turnstile), 2, 3
- ⊢_{DYN} (evaluation), 2, 9
- ⊢_{STAT} (elaboration), 2, 4
- ⇒, 1–3
- ⟨ ⟩ (options), 3, 35

A

'a, 36
a (see address)
 abs (absolute value), 54
 abstraction, 52, 85, 143
 abstype declaration, 21, 50, 52, 113, 132
 addition of numbers (+), 54
 address (or reference) (*a*), 14, 15, 23, 24
 (see also polymorphic reference)
 Addr_s (addresses), 23
 admissibility, 88–94, 104, 105, 118
 checking, 89, 92, 93, 103, 104, 144
 of elaboration tree, 93, 94, 97–99, 105, 109, 144
 of object or assembly, 93, 102, 117
 admissification, 100–103
 failure of, 103, 144
 admissifier, 100–102
 admit equality
 for types, 50
 for type functions, 50, 70
 for type names, 50, 68, 70
 ambiguities, 147, 148
 and, 28
 APPLIANCE, 26
 application
 of basic value (APPLY), 14, 16
 of (function) closure, 9–12, 16
 of value constructor, 16
 of exception name, 16
 of `ref`, 16
 of `:=`, 16
 application of functor (see functor application)
 applicative type variable, 43
 APPLY, 14, 16
 arity
 of type name, 47, 70, 103, 141, 144

 of type function, 48, 70, 103, 141, 144

assignment (`:=`), 16, 41, 45

B

b (basic value), 14
B (see basis)
*B*₀ (see initial basis)
 bare language, 7
 bare principality, 105–108, 110, 113, 114, 130
 BasExName (basic exception names), 14
 basic value (*b*), 14
 basis (*B*), 2
 static, 2, 4, 95
 dynamic, 2, 9, 28
 combined, 2, 4
 BasVal (basic values), 14
*B*_{DYN} (dynamic basis), 2, 9, 28
 binary trees, 47
 Bind (exception), 14
 body of type scheme, 37
 bool, 148
 bound datatype name, 112
*B*_{STAT} (static basis), 2, 4, 95

C

C (context), 2, 35, 41
 "cannot open *s*", 148
 cartesian product (\times), 13
 case, 38, 138
CE (constructor environment), 6, 48, 71, 89, 97, 144
 Clos (closure of types), 37, 39, 40, 55, 135, 138, 147
 closed functor, 77
 closed signature, 61
 closure (function value), 9–12, 16, 49
 closure of types etc (Clos), 37, 39, 40, 55, 135, 138, 147

closure rules (signatures and functors),
61, 77

coercion of numbers (**real**)

coercive matching (see signature ascription)

comments, 147

Complex, 57, 58, 72, 141

COMPLEX, 61, 62, 72, 141

COMPLEX_Pair, 63

con (value constructor), 11, 15, 16,
20, 21, 47, 49, 132–134

ConBind (constructor bindings), 147

conclusion (of a rule), 3

condesc (constructor description), 62

conformity, 30, 34, 84–86

“consing” an element to a list (*::*),
37, 45, 148

consistency, 47, 48, 88–93, 97, 100,
104, 105, 111, 117, 143, 144

constant, 48
special, 2, 49

constructor description, 62

constructor environment (*CE*), 6, 48,
71, 89, 97, 144

contents of (dereferencing), 41, 45

context (*C*), 2, 35, 41

contraction, 117, 120–123

contravariance and functors, 87

Core Language
static semantics, 35–55
dynamic semantics, 13–24

cover, 92, 109–111

cumulative interpretation, 3

curtailment, 28, 30–34, 67, 68, 71, 73,
83, 84, 88, 134, 140, 141

cycle-freedom, 88, 93, 105
checking, 93, 103, 104, 144, 145

D

dangling references, 23

datatype (see below)

datatype binding (*datbind*), 21, 147

datatype declaration, 6, 20, 21, 25,
47–52, 82, 113, 132, 135, 141,
147

datatype description (*datdesc*), 62, 96,
147

datatype specification, 91, 113, 126–
128, 132, 141

datbind (datatype binding), 21, 147

DatBind (datatype bindings), 147

datdesc (datatype description), 62, 96,
147

declaration, *dec* (Core), 6, 56

dependent type, 26

dereferencing (!), 41, 45

derived forms, 7, 148

determinacy, 23, 24

Dom (domain), 21

dynamic
semantics (Core), 13–24
semantics (Modules), 25–34
basis, 2, 9, 28

E

e (exception value), 18

\mathcal{E} (structural contraction), 117, 118,
120–123

E (see environment)

EE (see exception environment)

elaboration, 1, 4, 52
tree, 3, 135–137

else, 36, 148

en (exception name), 18, 19, 24

enrichment (\succ), 31, 71, 86, 87, 141

ens (exception name set), 19, 22, 43

environment (*E*)
static, 35, 57
dynamic, 9, 18, 27, 29–31, 42, 135

eqtype, 73, 79, 97, 113, 114, 125, 126,
141

equality, 74

equality function (dynamic semantics),
14, 15, 49

- equality (static semantics)
 - after signature ascription, 68
 - and realisation, 70
 - in functor result, 141
 - in **abstype** declarations, 50–52
 - in **datatype** declarations, 50–52
 - in **datatype** specifications, 103, 112–114, 126–128, 144
 - in **eqtype** specifications, 103, 113, 125, 126, 144
 - in **type** declarations, 50
 - in **type** specifications, 103, 113, 125, 144
 - equality of semantic objects, 13–15
 - equality attribute
 - of type name (see admit equality)
 - of type variable (see equality type variable)
 - equality type, 50, 68
 - equality type specification, 73, 79, 97, 113, 114, 125, 126, 141
 - equality type variable, 49, 50
 - equivalence of phrases, 31, 33, 62, 81, 83, 84, 142
 - evaluation, 1, 9, 21–24
 - tree, 3
 - exbind* (exception binding), 21, 132, 147
 - exception, 19, 21
 - exceptions, 18–21, 40, 43, 45, 50, 132
 - exception binding *exbind*, 21, 132, 147
 - exception constructor, 19, 132
 - exception constructor environment (see exception environment)
 - exception convention, 15, 19
 - exception declaration, 19, 21, 40, 45, 132, 147
 - exception description, 62, 118
 - exception environment (*EE*)
 - static, 35, 40, 45, 71
 - dynamic, 21, 148
 - exception name (*en*), 18, 19, 24
 - exception name set (*ens*), 19, 22, 43
 - exception packet, 4, 15, 18, 19
 - exception specification, 91, 132
 - exception value (*e*), 18
 - excon* (see exception constant or constructor), 19, 132
 - excons* (exception constructor set), 30
 - exdesc* (exception description), 62
 - execution, 1
 - exhaustive patterns, 16, 71
 - exn**, 18–20
 - exp* (expression), 8, 9, 12, 15, 17, 43
 - expansive expression, 45, 46, 138
 - expression (*exp*), 8, 9, 12, 15, 17, 43
 - expression row, 35
 - exprow* (expression row), 35
 - Extended ML, 86
 - extensible datatype, 19
 - ExVal (exception values), 18
- F**
- F* (functor environment), 4, 28, 33, 78, 81, 95
 - factorial, 36
 - FAIL (failure in pattern matching), 12, 18
 - false**, 148
 - $\overset{\text{fin}}{\mapsto}$ (finite map), 13, 35
 - Fin (finite subset), 13
 - finite map, 13, 35
 - fixed on *N*, 101–103, 118, 119, 129
 - flexible
 - structure name, 59, 61, 70, 82, 105
 - type constructor, 67, 114
 - type name, 59, 61, 70, 82, 105
 - fn**, 9, 38, 45
 - free name, 61, 105, 109, 130
 - free structure, 117
 - free type structure, 117
 - fresh
 - structure name, 56, 57, 78, 79, 82, 89

- type name, 47, 78, 79, 82
 - type variable, 53
 - FRONT_END, 65, 66
 - fun, 7, 38, 136
 - funbind* (functor binding), 33, 78–81, 83–86, 137
 - function (*fn match*), 9, 38, 45
 - function declaration (*fun*), 7, 38, 136
 - function type (\rightarrow), 36
 - function type expression (\rightarrow), 26, 41
 - function, user-defined, 1, 9, 10, 12, 16, 36
 - function-value binding (*fvalbind*), 7, 38, 136
 - functor, 32
 - functor, 32–34, 76–87
 - actual argument, 32
 - application, 32, 81–84, 134, 137
 - argument signature, 32, 78, 88, 112, 116
 - body, 32, 76–80, 82, 116
 - formal argument, 32
 - higher-order, 86, 87
 - result signature, 32, 33, 80, 81, 142
 - functor binding (*funbind*), 33, 78–81, 83–86, 137
 - functor closure, 32, 33, 137
 - functor declaration (*fundec*), 32, 33, 78–81, 134
 - functor environment (*F*), 4, 28, 33, 78, 81, 95
 - functor identifier (*funid*), 32, 33, 78, 81
 - functor signature (Φ), 76, 78, 80–83, 86, 92, 95, 141–143
 - functor signature expression (*funsigexp*), 77
 - functor signature matching, 86
 - functor specification (*funspec*), 77, 86, 134
 - fundec* (functor declaration), 32, 33, 78–81, 134
 - funid* (functor identifier), 32, 33, 78, 81
 - funsigexp* (functor signature expression), 77
 - funspec* (functor specification), 77, 86, 134
 - fvalbind* (function-value binding), 7, 38, 136
- ## G
- G* (see signature environment)
 - garbage collection, 22
 - generalisation (\succ), 8, 37, 46, 53, 55, 71, 72, 141
 - generative name, 78, 80, 82
 - generative signature expression, 26, 59, 90, 134
 - generative structure expression, 6, 29, 56, 57, 82, 89, 99, 134
 - grounded
 - basis, 102, 106, 110, 130
 - in *N*, 102, 103, 117–120, 146
- ## H
- handle, 19, 38, 40
 - hypothesis (of a rule), 3
- ## I
- I* (interface), 27, 30–33, 77, 135, 137
 - IB* (interface basis), 31–33
 - identifier (*id*)
 - long, 27, 71, 132
 - qualified, 57
 - status, 20, 21, 31, 132–134, 147
 - identity of values (see equality)
 - IE* (interface environment), 30
 - if, 36, 148
 - information hiding, 26, 52, 58, 82, 85
 - imperative type, 45, 53
 - imperative type variable, 43, 45, 46
 - implementation, 26, 52, 53, 100–104, 106, 111–113, 116, 120–130

in (injection), 4, 5, 8, 10
 in, 19, 27, 36, 40
 include, 31, 62, 96, 97, 120, 133, 135, 144
 incremental interpretation, 3, 36
 induction
 on the depth of inference, 21–24, 38, 99, 120
 structural, 21
 inference tree (∇), 3, 98, 99, 118
 inference rules
 static semantics (Core), 5, 8, 36, 39, 44, 135–137
 static semantics (Modules), 5, 60, 64, 95–104
 dynamic semantics (Core), 10, 15–17
 dynamic semantics (Modules), 10, 33
 initial basis
 static, 61, 109, 130
 dynamic, 14, 148
 injection (in), 4, 5, 8, 10
 input/output, 14, 148
 instance (\geq)
 of signature to signature, 71, 87
 of signature to structure, 70, 71, 87, 99, 105, 141
 of functor signature, 82, 87
 in matching, 72
 instream, 14
 int, 36
 Inter, 30, 31, 33
 interface (I), 27, 30–33, 77, 135, 137
 interface basis (IB), 31–33
 interface environment (IE), 30
 Interrupt (exception), 14, 24
 InvOrdItem, 57
 Io, 148
 it, 9, 148

K

kind, 26

L

lab (label), 35, 54, 137
 lamp, 25
 let
 expression (Core), 19, 27, 36, 40
 expression (Modules), 29
 lists, 37, 41, 45, 50
 list reversal, 45
 local
 declaration (Core), 52, 132
 declaration (Modules), 29, 132
 specification (Modules), 31, 62, 74
 long identifiers (e.g. *longexcon*), 35, 134

M

m (structure name), 6, 7, 56, 59, 63
M (structure name set), 61, 95
 macro, 82
 map, 42
 match (*match*), 9–12, 16, 38
 exhaustive, 16, 71
 in closure, 9–13
 Match, 13
 Match (exception), 14, 17
 matching
 a signature against a signature, 86, 87
 a structure against a signature, 26, 59, 67–70, 72, 73, 87, 116, 140
 via a realisation, 75, 81, 86, 87
 functor signatures, 86
 maximise equality
 type environments, 51, 112
 signatures, 108, 112, 114, 146
 maximum fixed point, 52, 112
mem (memory), 16, 22, 23, 42, 43
 memory typing, 42, 43
 mistakes, 147, 148
 modification (+)
 of finite maps, 35
 of environments, 36

Modules

- static semantics, 56–131
- dynamic semantics, 25–34
- monotype, 36, 45, 72
- monster signature, 88, 92, 108, 146
 - relative, 90, 112
- multiplication of numbers (*), 54

N

- n* (name, see structure name and type name)
- N* (name set), 4, 7, 61, 95, 118
- n*-tuple, 16
- name (see structure name, type name and exception name)
 - bound, 59, 61, 70, 80, 92, 98
 - free, 61, 105, 109, 130
- name set (*N*), 4, 7, 61, 95, 118
- names (free names), 4
- nil*, 148
- non-expansive expression, 43, 46, 138
- num*, 54

O

- occurrence, 117
- of (projection), 95
- of
 - in case expression, 38, 138
 - in constructor binding, 1
 - in exception binding, 42
- op*
 - on variable or constructor, 54
- open* (declaration), 7, 20, 25, 27, 30, 34, 84, 133
- open* (specification), 31, 62, 98, 133, 144
- options (*< >*), 3, 35
- OrdItem*, 57
- ostream*, 14
- overloading, 54

P

- p* (packet), 4, 15, 18, 19
- Pair*, 57, 58, 72, 141
- PAIR*, 61, 62, 72, 141
- packet (*p*), 4, 15, 18, 19
- parsing, 1, 132–134
- pat* (pattern), 12, 17, 18, 38, 132
- patrow* (pattern row), 54
- pattern, 12, 17, 18, 38, 132
- pattern matching, 12, 17, 18
 - with *ref*, 16
- pattern row, 54
- pervasives (see also initial basis), 54, 61
- phase distinction, 26
- polymorphic
 - functions, 37, 38, 46
 - references, 41, 46, 138, 139
 - exceptions, 42, 43, 46
- polymorphism, 38, 39, 71, 81, 82, 137, 138
- polytype, 36
- principal
 - admissifier, 100–103
 - environment, 52–55, 147
 - equality-, 68, 76, 78, 111–114, 145, 146
 - signature, 27, 68, 69, 76, 87, 94, 100, 105–131, 147
 - type, 27, 52–55, 147
- principality theorem, 55, 118
- printing, 69
- product type (*), 61
- program (*program*), 1–4
- projection (of), 95
- proof, 98, 99, 105, 118

Q

- qualified identifier, 57
- quantification (of type variables), 37, 39–46, 48, 138

R

- r (record), 13
- \mathcal{R} (rule set), 98
- raise, 19
- real (type), 2
- real constant, 2
- realisation (φ), 70, 72, 98–100, 102, 117, 118, 141
 - inverse, 93, 109
 - uniqueness of, 72, 74, 75, 81, 83, 87, 141
- rec, 10, 12, 38, 136
- Rec (recursion operator), 9–12
- record
 - r , 13
 - type (ρ), 35, 54
- Record (records), 13
- recursive function (see also fun), 9–12, 38
- ref
 - the type name, 41
 - the value constructor, 14–16, 41–46
- reference (see address)
- renaming of bound names, 82, 98
- respect equality
 - type environments, 51
 - signatures, 107, 108, 111–113, 146
- restrictions
 - closure rules, 61, 77
- rigid
 - basis, 79, 95, 106, 110, 130
 - structure name, 78, 95, 144
 - type name, 48, 59, 78, 144
- \mathcal{R}_{sig} , 96, 99, 104, 118
- rule
 - instance, 2
 - liberal, 96
- rule 1**, 137
- rule 2**, 7, 44
- rule 3**, 7, 9, 136
- rule 5**, 137
- rule 6**, 36, 44
- rule 7**, 7
- rule 8**, 137
- rule 9**, 7, 9, 136, 137
- rule 10**, 7, 9, 44
- rule 14**, 7, 136
- rule 15**, 7, 136
- rule 16**, 7, 136
- rule 17**, 7, 8, 39–41, 44, 136, 138, 139
- rule 19**, 7, 21, 49, 135
- rule 20**, 21, 49, 52
- rule 23**, 7
- rule 26**, 7, 8, 44
- rule 27**, 7, 36, 38, 136
- rule 28**, 48
- rule 29**, 7, 48, 135
- rule 30**, 7, 135
- rule 31**, 45
- rule 35**, 7, 136
- rule 36**, 7
- rule 42**, 7, 136
- rule 43**, 7
- rules 47–52**, 96, 118, 120
- rule 47**, 118
- rule 49**, 7, 118, 135
- rule 53**, 5, 60, 89, 99
- rule 54**, 142
- rule 55**, 81, 84, 143
- rule 56**, 142
- rule 57**, 5, 7, 8, 55, 60
- rule 58**, 5, 60, 142
- rule 61**, 5, 60, 136
- rule 62**, 5, 31, 60, 73, 84, 141–143
- rule 63**, 64, 90, 96, 120, 124
- rule 64**, 96, 120, 125
- rule 65**, 64, 83, 92, 104, 109–114
- rule 66**, 64
- rule 68**, 64
- rule 69**, 64
- rules 70–90**, 96, 120
- rule 70**, 118
- rule 71**, 125
- rule 72**, 97, 125, 126

rule 73, 21, 126–128
rules 74–78, 118
rule 74, 21, 118
rule 78, 118
rule 79, 96
rules 80–82, 118
rules 82–87, 62
rule 83, 96, 97
rule 84, 96
rules 86–87, 118
rule 86, 147
rules 88–89, 96, 97, 128–130
rule 90, 118
rule 95, 77
rule 96, 78, 80, 142
rule 99, 73, 78, 80, 83, 84, 141, 143,
 147
rules 100–102, 45
rule 100, 5, 7, 8, 60
rule 101, 64
rule 105, 11, 19, 21
rule 109, 11
rule 111, 11
rule 112, 11, 15, 16, 22
rule 113, 16
rule 114, 16, 22
rule 115, 16, 22
rule 116, 16
rule 117, 16, 21
rule 118, 16, 21
rule 120, 19
rule 121, 19
rule 123, 9, 10
rule 124, 12
rule 125, 12
rule 126, 12
rule 129, 10–12, 21, 28
rule 130, 19, 21
rule 131, 28
rule 133, 28
rule 134, 28
rule 135, 11, 12, 28
rule 136, 10

rule 137, 10
rule 138, 19, 22
rules 140–159, 12, 18
rule 143, 10–12
rule 153, 10–12
rule 156, 18
rule 157, 18
rule 158, 16, 18, 22, 23
rule 160, 10, 29
rule 161, 29
rule 162, 33, 137
rule 163, 29
rule 164, 10–12, 29
rule 165, 10, 28, 29
rule 166, 28, 29
rule 167, 28, 29
rule 168, 10, 28, 29
rule 169, 10, 28–30, 137
rules 170–186, 31
rules 178–180, 137
rule 171, 135
rule 187, 137
rule 191, 10–12
rule 194, 22, 28
rule 195, 22
rule 196, 22, 95
 run-time (see evaluation)

S

s (state), 2, 21–24, 43
S (structure), 6, 25–28, 56–59
scon (special constant), 2, 49
 scope (see also identifier status), 35,
 132
 of explicit type variable, 35, 40,
 41
 static, 32, 34–36, 76
SE (structure environment)
 static, 5, 6, 35, 57
 dynamic, 27, 29, 30
 semantic object, 3, 4, 13
 sentence, 2, 3

- separate compilation, 61, 77
- sequential
 - declaration (Core), 132
 - signature declaration, 61
 - specification, 61
 - structure-level declaration, 29
- shape, 27, 30, 71
- shareq* (sharing equation), 96, 97, 119
- sharing, 63–67, 70, 88, 90, 91, 100, 105, 116
 - between functor argument and result, 80–82
 - by parameterisation, 65
 - external, 61, 65
 - in dynamic semantics, 31
 - internal, 61, 65
 - of values, 66
 - of structures, 63, 90, 128
 - of types, 48, 63, 91, 113, 114, 130
 - specification, 63, 100
- sharing
- sharing equation, 96, 97, 119
- side-condition, 3, 98
- side-effect, 15–17
- sig**, 59, 90, 96, 124
- sigbind* (signature binding), 67
- sigdec* (signature declaration), 67
- sigexp* (signature expression), 59, 69, 95–100, 105–115, 118, 130
- sigid* (signature identifier), 59, 61, 96, 125, 134
- signature (Σ), 59–63
- signature**, 61
- signature ascription
 - coercive (curtailment), 28, 30–34, 67, 68, 71, 73, 83, 84, 88, 134, 140, 141
 - non-coercive (conformity test), 30, 34, 84–86
 - abstracting, 85, 86
- signature binding (*sigbind*), 67
- signature checker, 118
- signature environment (*G*)
 - dynamic, 31
 - static, 4, 95
- signature expression (*sigexp*), 59, 69, 95–100, 105–115, 118, 130
- signature identifier, 59, 61, 96, 125, 134
- signature instantiation (see instance)
- signature matching (see matching)
- soundness of type inference, 24, 37, 39–44, 138, 139
- spec* (specification), 59, 95–98, 118
- special constant (*scon*), 2, 49
- special value (*sv*), 49
- specification (*spec*), 59, 95–98, 118
- stamp (see structure name and type name)
- state (*s*), 2, 21–24, 43
- state convention, 15
- state typing, 43
- static
 - basis (basis), 2, 4, 95
 - semantics (Core), 35–55
 - semantics (Modules), 56–131
- status of identifier, 20, 21, 31, 132–134, 147
- status map (see also status of identifier), 132–134
- std_in**, 14
- std_out**, 14
- store (see state and memory)
- strbind* (structure binding), 6, 29, 73, 137
- strdec* (structure-level declaration), 29, 57
- strdesc* (structure description), 62, 118
- stream (input/output), 14
- strexpr* (structure expression), 6, 28, 57, 69
- strid* (structure identifier), 27, 56, 134
- string**, 3
- string constant, 3
- struct**, 6, 29, 56, 57, 82, 89, 99, 134
- structural contraction, 117, 118, 120–

123
 structure (S or (m, E)), 6, 25–28, 56–59
structure, 1, 29
 structure binding (*strbind*), 6, 29, 73, 137
 structure declaration, 1, 25, 28, 29, 56, 133
 structure description (*strdesc*), 62, 118
 structure environment (*SE*)
 static, 5, 6, 35, 57
 dynamic, 27, 29, 30
 structure expression (*strexpr*), 6, 28, 57, 69
 structure identifier (*strid*), 27, 56, 134
 structure-level declaration (*strdec*), 29, 57
 structure name (m), 6, 7, 56, 59, 63
 structure name set (M), 61, 95
 structure realisation (φ_{Str}), 70
 structure specification, 61, 133
 substitution, 53, 100
 substructure, 57
 subtraction of numbers ($-$), 54
 Supp (support), 70, 98, 109
sv (special value), 49

T

t (type name), 6, 7, 47, 48
T (type name set), 35, 48, 49, 61
TE (type environment), 6, 35, 48, 75, 138
then, 36, 148
topdec (top-level declaration), 1–12, 106, 130
 top-level declaration, 1–12, 106, 130
true, 148
 tuple, 16
 tuple type, 61
ty (type expression), 118–120
tycon (type constructor), 47
 tynames (free type names), 117

typbind (type binding), 48
typdesc (type description), 62, 96, 147
 type (τ), 36
type, 47, 48, 50, 66
 type (function on special constants), 2
 type abbreviations in signatures, 66, 67, 102, 114
 type binding, 48
 type-checking, 52, 53, 55
 type constraint ($:$), 40, 41
 type constructor (*tycon*), 47
 type constructor name (see type name)
 type declaration, 47, 48, 50
 type description (*typdesc*), 62, 96, 147
 type environment (*TE*), 6, 35, 48, 75, 138
 type explication, 74, 75, 83, 87, 107, 109, 112, 113, 142
 type-explicit signature (see type explication)
 type expression, 118–120
 type-expression row (*tyrow*), 118
 type function (θ), 47, 48, 62, 66, 140
 type name (t), 6, 7, 47, 48
 type name set (T), 35, 48, 49, 61
 type realisation (φ_{Ty}), 70, 111
 type scheme (σ), 6, 36–40
 closed, 37
 general, 37, 39, 40
 simple, 37–40
 type specification, 26, 125
 type structure (θ, CE), 6, 47–49, 89, 97, 138, 145
 type variable (*tyvar*, α), 36
 bound, 37
 equality, 49, 50
 imperative, 43, 45, 46
 applicative, 43
 explicit, 40, 41, 53, 138
tyrow (type-expression row), 118
tyvar (see type variable)
 tyvars (free type variables), 39

tyvarseq (type variable sequence), 147

U

U (explicit type variables), 35, 39–41, 53

unification

of types, 53

second-order, 66, 75, 92, 101, 114, 119

of structures and type structures
(see admissification)

union (\cup), 13

unguarded type variable, 41

V

v (value), 13–24

val, 1, 39, 132

valbind (value binding), 38

simple, 1

recursive, 9–12, 36

valdesc (value description), 46, 62, 118

value binding (*valbind*), 38

simple, 1

recursive, 9–12, 36

value constant (*con*), 11, 15, 16, 20, 21, 47, 49, 132–134

value constructor (*con*), 11, 15, 16, 20, 21, 47, 49, 132–134

value declaration (**val**), 1, 39, 132

value description (*valdesc*), 46, 62, 118

value variable (*var*), 20, 21, 132–134

value specification, 91, 132

var (value variable), 20, 21, 132–134

variable (see value variable)

variable environment (*VE*)

static, 6, 7, 35, 38, 110

dynamic, 10–12, 21

vars (set of value variables), 30

VE (see variable environment)

VE_{datbind}, 21

via φ , 75, 81, 86, 87

view of a structure (see also signature ascription), 28, 67, 73, 81, 88

VIEW1 and **VIEW2**, 67, 71, 140

W

well-formedness, 88, 92, 93, 105, 111

checking, 92, 104, 113

of functor signature, 92

of signature, 75, 106, 107, 110

of type structure, 145

wildcard pattern ($_$), 18

wildcard pattern row (\dots), 54

with, 21, 50, 52, 113, 132

Y

Yield, 98