Notation

This glossary sets out some of the mathematical notation in this book. Descriptions marked thus\(^\dagger\) have a reference in the main index. For notation, such as \textit{initials}(P) and \textit{Chaos}, with an obvious alphabetical place in the index, you should refer directly there.

Many pieces of notation whose use is relatively localised (to a single chapter or section) are not included below. An example is the notation associated with Sect. 17.4 on buffer tolerance.

**Sets and Numbers**

\begin{align*}
a \in x & \quad \text{set membership (true iff } a \text{ is in } x) \\
x \subseteq y & \quad \text{subset (}\forall a. a \in x \Rightarrow a \in y) \\
\emptyset & \quad \text{the empty set} \\
\{a_1, \ldots, a_n\} & \quad \text{set containing these elements} \\
x \cup y, \bigcup X & \quad \text{union} \\
x \cap y, \bigcap X (X \neq \emptyset) & \quad \text{intersection} \\
x \setminus y & \quad \text{difference (}= \{a \in x \mid a \notin y\}) \\
P(x) & \quad \text{powerset (}= \{y \mid y \subseteq x\}) \\
x \times y & \quad \text{Cartesian product (}= \{(a, b) \mid a \in x \land b \in y\}) \\
x \to y & \quad \text{the space of all functions from } x \text{ to } y \\
N & \quad \text{natural numbers (}\{0, 1, 2, \ldots\}) \\
Z & \quad \text{integers (}\{\ldots, -2, -1, 0, 1, 2, \ldots\}) \\
R & \quad \text{real numbers} \\
R^+ & \quad \text{non-negative real numbers} \\
\oplus, \ominus & \quad \text{addition and subtraction \textit{modulo} the appropriate base}
\end{align*}

**Logic**

\begin{align*}
x \land y & \quad \text{conjunction (} x \text{ and } y) \\
x \lor y & \quad \text{disjunction (} x \text{ or } y\)
\end{align*}
Notation

\(\neg x\) negation (not \(x\))
\(x \Rightarrow y\) implication (\(\equiv (\neg x \vee y)\))
\(x \Leftrightarrow y\) double implication (\((x \Rightarrow y) \land (y \Rightarrow x)\))
\(\forall x.\chi\) universal quantification (\(\chi\) holds for all \(x\))
\(\exists x.\chi\) existential quantification (\(\chi\) holds for at least one \(x\))

For LTL notation see p. 381.

Partial Orders

\([\sqcap X\) least upper bound
\([\sqcap X\) greatest lower bound
\(\mu f\) least fixed point of \(f\)

Communications

\(\Sigma\) (Sigma\(^I\)): alphabet of all communications
\(\check{\ }\) (tick) termination\(^I\) signal
\(\tau\) (tau\(^I\)): the invisible action
\(\Sigma^{\check{\ }\tau}\) \(\Sigma \cup \{\check{\ }, \tau\}\)
\(a.b.c\) compound event (see p. 14)
\(c\check{\ }x\) input\(^I\)
\(c!e\) output\(^I\)
\(|\{a, b\}|\) events associated with channels (see p. 15)

Sequence/Trace Notation (See pp. 30 and 36)

\(A^*\) set of all finite sequences over \(A\)
\(A^{\check{\ }*}\) \(A^* \cup \{s^\check{\ }\} | s \in A^*\)
\(A^{\omega}\) set of all infinite sequences over \(A\)
\(\langle\rangle\) the empty sequence
\(\langle a_1, \ldots, a_n\rangle\) the sequence containing \(a_1, \ldots, a_n\) in that order
\(s^\check{\ }t\) concatenation of two sequences
\(s \setminus X\) hiding: all members of \(X\) deleted from \(s\)
\(s \downarrow X\) restriction: \(s \setminus (\Sigma^{\check{\ }\} \setminus X)\)
\(#s\) length of \(s\)
\(s \downarrow a\) (\(a\) an event) number of \(a\)'s: \(#(s \downarrow \{a\})\)
\(s \downarrow c\) (\(c\) a channel) sequence of values communicated on \(c\) in \(s\)
\(s \preceq t\) (\(\equiv \exists u.s^\check{\ }u = t\)) prefix order
Transition Systems (See Sect. 9.1)

\[ s \parallel t \ (\subseteq \Sigma^{*\tau}) \text{ generalised parallel} \]
\[ s \parallel t \ (\subseteq \Sigma^{*\tau}) \text{ interleaving} \]
\[ \overline{S} \text{ closure of } S = S \cup \{ u \in \Sigma^{au} | \forall s < u.s \in S \} \]

Note that sequence-like notation is also used to denote vectors indexed by arbitrary sets, usually with reference to mutual recursion, for example \( \langle B_s^\infty | s \in T^* \rangle \).

Transition Systems (See Sect. 9.1)

\[ \hat{C} \text{ The set of nodes in transition system } C \]
\[ P \xrightarrow{a} Q \ (a \in \Sigma^{*\tau}) \text{ single action transition} \]
\[ P \xrightarrow{s} Q \ (s \in \Sigma^{*\tau}) \text{ multiple action transition with } \tau \text{'s removed} \]
\[ P \xrightarrow{t} Q \ (t \in (\Sigma^*)^{*\tau}) \text{ multiple action transition with } \tau \text{'s retained} \]
\[ \tau^*(P) = (\{ Q | P \xrightarrow{\tilde{0}} Q \}) \text{ } \tau \text{-expansion of } P \text{ (see p. 358)} \]
\[ P \text{ ref } B \text{ } P \text{ refuses } B \]
\[ P \uparrow \text{ } P \text{ diverges} \]

Processes

The syntax of CSP\(_M\) is set out in the documentation for FDR that can be found on this book’s web-site. There is a slightly out of date version in Appendix B of TPC.

\[ \mu p.P \text{ recursion} \]
\[ a \rightarrow P \text{ prefixing} \]
\[ ?x : A \rightarrow P \text{ prefix choice} \]
\[ (a \rightarrow P \mid b \rightarrow Q) \text{ guarded alternative} \]
\[ P \sqcap Q \text{ external choice} \]
\[ P \sqcup Q, \sqcap S \text{ nondeterministic choice} \]
\[ P > Q \text{ sliding choice} \]
\[ P \* Q \text{ conditional choice} \]
\[ b \& P \text{ conditional guard (see p. 14)} \]
\[ P \parallel Q \text{ synchronous parallel} \]
\[ P X || Y Q \text{ alphabetised parallel} \]
\[ P \parallel Q \text{ generalised parallel} \]
\[ P \parallel Q \text{ interleaving} \]
\[ P \setminus X \text{ hiding} \]
\[ f[P] \text{ renaming} \text{ (functional)} \]
\[ P[R] \text{ renaming} \text{ (relational)} \]
\[ P[a/b] \text{ renaming} \text{ (relational, by substitution)} \]
\[ P; Q \text{ sequential composition} \]
\[ P[a \leftrightarrow b]Q \text{ link parallel} \]
Notation

\[ P \gg Q \] piping \(^I\) (or chaining)
\[ P \mathbin{\|} Q \] enslavement \(^I\)
\[ P \vartriangle Q \] interrupt \(^I\)
\[ P \Theta_A Q \] throw \(^I\)
\[ P[x/y] \] substitution (for a free identifier \( x \))
\[ P/s \] ‘after’ \(^I\) operator
\[ P \downarrow n \] restriction to depth \( n \) (model dependent)
\[ L_H(P) \] lazy abstraction \(^I\)
\[ \tau P \] \( P \) “prefixed by” a single \( \tau \) action (equivalent to \( P \))
\[ fv(P) \] \( P \)’s free variables/identifiers

Semantic Models

\( T \) traces model \(^I\)
\( N \) failures/divergences model \(^I\)
\( F \) stable failures model \(^I\)
\( R \) stable revivals model \(^I\)
\( A \) stable acceptances (ready sets) model \(^I\)
\( RT \) stable refusal testing model \(^I\)
\( FL \) linear behaviours (acceptance traces) model \(^I\)
\( M \downarrow \) finitary divergence-strict extension of \( M \)
\( M \downarrow^\omega \) infinitary divergence-strict extension of \( M \)
\( M^\sharp \) “seeing past divergence” extension of \( M \)
\( \subseteq_T \) traces refinement
\( \subseteq_{FD} \) failures/divergences refinement
\( \subseteq_F \) failures refinement (i.e., over \( F \))
\( \subseteq_{FL} \) refinement over \( FL \)
\( \subseteq \) refinement over whatever model is clear from the context
\[ P \leq Q \] strong order \(^I\) (over divergence-strict models)
References

68. Hyman, H.: Comments on a problem in concurrent programming control. CACM 9, 1 (1966)
89. Lowe, G.: On CSP refinement tests that run multiple copies of a process. ENTCS 250, 1 (2009)
Index

A
Abstraction, 104, 125, 176, 413, 414
Acceptance (ready set), 162, 174, 199, 256, 258, 260, 262, 358, 359, 361
minimal, 175, 314
saturated collection of, 314
Acceptances (ready sets) model
stable (A), 260–262, 265, 266, 270, 271, 282, 310, 312, 358, 362
Acknowledgement messages, 87
ACP, 293
’after’ operator (P/s), 40
Algebraic operational semantics (AOS), 294, 299–305
Algebraic semantics, 23–29, 33, 189, 236, 293–317
Alphabet, 3
explicit vs implicit, 49
of a process, 49, 65, 66
transformation, see renaming
Alphabetised parallel (X ∥ Y), 49–57, 97, 104
as conjunction of trace specifications, 61–64
in terms of ∥, 59, 61
indexed, 51
laws of, 50, 51
termination of, 136
traces of, 53
with intrinsic alphabets, 65
Alphabetised process, 177, 183
Alternating bit protocol (ABP), 87–91, 93, 112, 127, 410
may diverge, 99
AOS form, 294, 295–305
Armstrong, Philip, xii, 184, 446
assert, 144
Assignment, 9, 135, 423, 426, 465, 466
Asynchronous systems, 407, 420, 452
Atomic equivalent program, 465–469, 471, 476, 479

B
Bag, 58, 381, 406
simplified, 440, 450, 451
tolerant of dirty variables, 449
Barrett, Geoff, 184, 186, 279
Barrier synchronisation, 492
Behavioural model, 23, 42, 219, 229, 255–270
Binary futures model, 268
Bird, Richard, 150
Bisimulation, 194–200, 378, 498
divergence respecting weak (DRW), 196–199, 257, 364, 365, 369
maximal, 195, 197, 200, 358, 364
strong, 174, 195, 196, 199, 214, 217, 358, 361, 364, 369
and divergence, 198
weak, 196, 197
Breadth first search (BFS), 162, 166, 167, 374, 375
Brookes, Stephen, xi, 227, 267, 317
Broy, Manfred, 291
Buffer, 8, 94, 97, 102, 110, 125, 280, 381, 406–412, 415, 497
failures specification of, 121, 122
failures-divergences specification of, 122
infinite, 8
law 1 (BL1), 110, 122
law 5 (BL5), 110, 122

Buffer (cont.)
  law 6 (BL6), 410
  specification is infinite state, 21
  specification of, 12
  trace specification of, 37, 39
  weak, 405
Buffer tolerance, 407–412
  left, 409
  notation, 409
  right, 409
  strong, 409
  weak, 409
Bulk synchronous processing (BSP), 496
Bully algorithm, 327–339
Burglar alarm, 491
Busy wait, 78, 99, 286, 438
Butler, 55, 443

C
Casper (security protocol analyser), 398
CCS, 194, 220, 228, 253, 265, 497, 499
CEGAR, 184, 413–415
Channel, 7, 14, 15, 55, 241, 497
  array of, 14, 55
  internal and external, 51, 93
Chaos, 16, 36, 122, 238, 240, 261, 294, 321
ChaosA, 126
Characteristic process, 39, 56, 121, 281
  chase, 343, 370–373, 378, 496
Checker, 150
Choice
  angelic, 219, 220
  conditional, 12–14, 20
    for collapsing case definitions, 13
    laws of, 24–28
  external (□), 10, 11, 117, 487
    ambiguous initial events, 10
    infinite, 208
    laws of, 24–28, 102, 106, 261, 302, 303
    operational semantics of, 204, 206
  guarded, see guarded alternative
  nondeterministic (Γ), 11, 12, 208
    infinite, 226
    laws of, 24–28, 310
    operational semantics of, 203, 207
  nondeterministic (Γ), 117
    prioritised external, 487
    sliding (>), 96, 299
    laws of, 301, 307
    operational semantics of, 206
Clean variable, 447
Closed covering principle (CC), 277, 278
Closed CSP term, 202
Closed process, 278, 280
Closed property, 250, 379
Closure of a set of traces, 223, 278
Combinator operational semantics, 201,
  204–221, 263, 347
  CSP simulation of, 214–219
Comms(e), 202, 208
Communication graph, 51, 54, 76
Communications, 3, 18, 19
  agreement with environment on, 193
  are instantaneous, 3
  multi-part, 7, 14
Communications protocols, 86–91
Compiler
  of CSPM in FDR, 148, 155, 161
  written in CSPM, 421–430
Complete partial order (cpo), 273, 280, 287
Complexity of deciding refinement, 362
Compositional development, 39
Compression, 97, 104, 173–184, 186, 231,
  240, 363–373, 378, 419, 450
  in SVA, 430–434
Concurrent condition, 177
Concurrency, 45
Conflict
  read/write, 449
  write/write, 449
Confluence, 248–250, 252, 253, 371, 372, 411,
  412
  defined semantically, 251
Congruence, 230, 231, 258, 290, 293
  as distinct from a model, 232
  congruence theorem, 229
Constant symbol, 393, 399, 400
Constraints via parallel composition, 62, 63
Constructiveness, 32, 47, 100–102, 133,
  236–238, 243, 250, 280
Contexts, 39, 230, 251
Continuation, 426, 427
Continuity, 233–236
Contraction mapping, 234
Controlled error model, 88, 100, 126, 379
COPY, 7, 8, 16, 37, 41, 61, 109, 209, 210, 324,
  391, 394
COPY(a, b), 51, 93, 97
COUNT, 6, 7, 15, 21, 32, 37, 48, 57, 58, 103,
  108, 133
Covert channel, 127, 128
Creese, Sadie, 399, 415
Cryptographic protocol, 371, 397
CSP as a specification language, 17, 18
CSP-like operator, 208, 214, 218
Currie, Andrew, 381
Cyclic communication network (CCN), 249,
D
Data independence, 184, 390–406, 415, 453, 454
extended by arrays, 398
Data refinement, 457
Data-independent induction (DII), 398–406
Data-type, 7, 14, 153, 422
Davies, Jim, 355
De Nicola, Rocco, xi
Deadlock, 19, 54, 56, 73–86, 97, 98, 104, 124, 138, 144, 177, 178, 193, 230, 236, 264, 319, 442, 443
checking in FDR, 147
in CSPM, 129
specification of freedom from, 117, 121, 240
specified over A, 261
Deadlock graph, 76
Deadlock Rule 1, 77
extension to, 77, 78
Declarative semantics, 9, 132, 134, 191
Dekker’s algorithm, 441, 450
delay, 343
Denotational semantics, 31, 115, 189, 229–253
Depth first search (DFS), 122, 167, 169, 374, 378
Determinism, 11, 120, 121, 144, 145, 171, 174, 243, 247–251, 252, 274, 369, 370, 496
and testability, 120
checking for, 170–173
checking in FDR, 147
checking over F, 173
corresponds to maximality over N, 247
extensional vs intensional, 247
in CSPM, 129
over F, 171
Deterministic refinement, calculating, 170
DF, 121
diamond, 174, 175, 367–369, 378, 415
Dining philosophers, 53–55, 56, 58, 76, 102, 103, 167, 176, 178, 179, 183, 452
asymmetric, 55
compression of, 176
using shared variables, 441–445
Dirty variable, 447–451, 454, 467
Distributed algorithm, 327
Distributed termination, 136, 137, 228
operational semantics of, 207
Distributivity (over □), 25, 48, 233, 255, 299
div, 16, 118, 119, 122, 131, 147, 182, 257, 196, 203, 238, 273, 283, 294, 296, 321, 359
and finite-behaviour models, 237
is bottom of divergence-strict model, 242
checking in FDR, 147
in CSPM, 129
non-strict, 283–289
proving absence of, 124, 125
Divergence strictness, 118, 139, 181, 182, 229, 241, 271–281, 283, 299, 461
models without, 229
weak (WDS), 285, 286
Divergence-strict model, 252, 282, 365
Divergences
strictness after, 242
Divergences strictness, 223
divergences (P), 241
calculation of, 244–246
done, 72, 169, 481

E
Enslavement (\//), 110, 135, 497
Environment (external), 3, 11, 138, 192, 219, 326, 458
Environment (semantic), 202
Equivalence relation, 199
Error handling, see controlled error model
Error-strictness, 461, 463
Events, 20
events(c), 15
Evolving state, 325
Expressiveness of CSP, 208, 214–219
Extensionality, 41, 247

F
Factoring an LTS, 196, 198, 364, 369
Failures, 115–123, 137, 138, 145, 237
calculation of, 238
channel based, 241
inadequacy of, 123
stable, 241
Failures model
infinitary divergence-strict (F\lim\omega), 280, 283
seeing beyond divergence (F^\omega), 286, 358
as proxy for N, 239
inadequacy of, 118
failures(P), 115, 222
failures \_\_ (P), 222, 241
  calculation of, 244–246
Fairness, 99, 276, 442, 443
Fault tolerance, 126, 128, 172, 327
  availability, 184
  batch mode, 149
  calculation of bisimulation, 364
  compaction of data structures, 149
  compilation, 161
  control of, 148
  debugger, 68, 70, 144, 146, 147, 162, 163, 169, 170, 388
  DFS search, 167
    via divergence checking, 168
  divergence checking, 169
  evaluate mode, 144
  graph mode, 144, 175
  GUI, 144
  implementation of recursion, 203
  interrupting, 148, 149
  memory control (FDRPAGEDIRS), 149
  memory management, 167, 373–378
  multiple counter-examples, 148, 160, 342
  parallelisation of, 376–378
  Status window, 144, 148
  two-level compilation, 161
Finite output property (FOP), 411
Finite program, 294, 295, 297, 309
Finite state-process, 357
Finite-behaviour model, 234, 237, 255–270
  principal, 266
  Finite-state machine, 194
  Finite-state process, 21, 43, 119, 122
  Fitzgerald, John, 480
Fixed point, 31, 100, 231, 238, 271, 273, 278–280
  for mutual recursions, 32
    reflected, 286–289
  Fixed-point induction, 250, 251
  FD*-behaviour, 272
  FL*-behaviour, 271, 272, 283
  FL-behaviour, 256, 258
  Forgetful renaming, 104, 412, 413
  FORK, 53
  FORKS, 58
  Formal Systems, 184
  Format, 205, 207, 212, 214, 216
  Full abstraction, 230, 251, 252, 270
  Function, 105
    injective, 103
  Functional process, 411, 412
  Functional programming, 144, 150, 191
    with lists, 151
    with sets, 152
  Functional programming language, 9, 191
G
  Gardiner, Paul, 184
  Generalised labelled transition system (GLTS), 174, 199, 359, 367
  Generalised parallel (\[\]), 59–61
    laws of, 59
    traces of, 60
  Generalised parallel (\[ [X]\])
    laws of, 97, 104
  Goldsmith, Michael, 184, 415
  Greatest lower bound, 242
  Guarded alternative, 5–7, 8, 10, 20
    representable using \[\[\], 10
  Guardedness, see constructiveness
  GUI of SVA, 436, 437
H
  Halting problem, 131
  Hamiltonian circuit, 159, 160, 162, 163
  Handshaking, 3, 18, 45, 61, 407
  Hash table, 374
  Haskell, 9, 150
  He Jifeng, 317
  Head normal form, 295
  Healthiness conditions, 236, 237, 251, 276, 277
    for divergence-strict models, 274
    for \[\], 284
    of A, 260
    of binary futures model, 268
    of \[\], 236
    of \[\], 257
    of \[\], 272
    of N, 242
    of R, 264
    of R\, 262
    of T, 236
  Heather, James, 184
  Hennessy, Matthew, xi
Index 523

Hiding (\(\backslash\ X\)), 93–102, 112, 176
\(\bigvee\) never hidden, 137
can cause divergence, 99
discontinuous over \(\sqsubseteq_T\), 233
in compression, 177
infinite, 99, 226
laws of, 94–97, 104, 300
negates guardedness, 100
operational semantics of, 206
traces of, 97

Hoare, C.A.R. (Tony), v, xi, 13, 37, 65, 152, 253, 260, 264, 317

Hopkins, David, 446, 480

Hyman’s algorithm, 420, 450, 451

I
Identifiers
non-process, 12, 31, 201
binding of, 9, 28
scope of, 132
process, 4, 203
Idling state, 324, 325
Imperative programming, 135, 185
include, 177
Induction, 115, 124, 179, 398, 415
Inductive compression, 179, 181, 183
Inference system, 371, 398
\(\text{infinities}(P)\), 223
Infix dot, 7, 14
\(\text{initials}(P)\), 40
Inmos, 185
Input, 7–9, 15, 324, 326
Intensionality, 41, 247
Interaction, 45
Interface parallel, see generalised parallel
Interleaving (\(\parallel\)), 57, 59
as abbreviation for disjoint parallel, 58
in terms of \(\parallel\ X\), 59
laws of, 57
operational semantics of, 212
recursion through, 57
traces of, 58
Internal choice, see choice, nondeterministic (\(\gamma\))
Interrupt (\(\bigtriangleup\)), 138, 139, 141
laws of, 139, 304
operational semantics of, 207
Intruder process, 371, 372, 398
Invariant, 180–182, 183
Iteration, sequential, 134

J
Jackson, David, 184, 355
Jagadeeshan, Lalita, 253

K
Kannellakis, Paris, 362
Knight’s tour, 481–486, 489, 497, 499, 500
Knuth’s mutual exclusion algorithm, 451
König’s Lemma, 224, 272, 277

L
deterministic, 116, 247
factoring by an equivalence relation, 414
finitely branching, 223
Lamport, Leslie, 439, 471
Latency, 407
Law, 23, 24, 28, 33, 43, 94, 132, 257, 293, 295, 299, 310, 317
associative, 23, 50
commutative, 23
distributive, 23
idempotence, 23
slide, 300, 301
step, 27, 95–97, 299, 300
symmetry, 23, 50
unit, 23
zero, 23
Lazić, Ranko, 172, 184, 191, 379, 380, 393, 397
Lazy abstraction, 125–128, 240, 281, 286, 379
does not introduce divergence, 126
Lazy compression, 370–373
Lazy independence, 127, 275
lazynorm, 363, 370, 373, 381
Leadership election, 327
Leaf compression, 178, 182, 373
Least fixed point, 280, 288
Left value, 426
Leuschel, Michael, 381
Levy, Paul, 283, 290
Linear behaviours (acceptance traces) model
divergence-strict (\(\mathcal{F}\mathcal{L}^0\)), 272–275, 282
infinite divgence-strict (\(\mathcal{F}\mathcal{L}^{\infty}\)), 282
seeing beyond divergence (\(\mathcal{F}\mathcal{L}^\gamma\)), 289, 309
stable (\(\mathcal{F}\mathcal{L}\)), 256–262, 266, 269, 270, 310, 312
Linear behaviours (acceptance traces) model
stable (\(\mathcal{F}\mathcal{L}\)), 271, 272, 274, 282
Linear observation, 26, 255, 256, 258, 267, 268
Linear observations (acceptance traces) model
divergence-strict (\(\mathcal{F}\mathcal{L}^0\)), 310
Linear order, 473–480
Linear temporal logic (LTL), 438, 471
Link parallel, 109–112, 114
Liveloak, see divergence
Liveness, see specification, liveness, 343 in SVA, 442
LNF (linear normal form), 307–313, 315
Local search, 375
Local variable, 450
Logical relations, 392
Lowe, Gavin, 380–382, 398
LTL (linear temporal logic), 381–383, 442

M
Machine readable CSP_M, 9, 12, 15, 19–21, 64, 65, 105, 112–114, 129, 141, 142, 144–157, 191, 421, 446
type checking, 150
Machine readable (CSP_M), 175
Mad postman routing (virtual network), 84–86
Many-way synchronisation, 61, 68, 72, 220, 485, 499
Massart, Thierry, 381
Maximal progress, 339, 340, 496
Mergesort, 111
Metric space, 229, 234, 239, 250, 380
Meyer, Albert, 253
Milner, Robin, xi, 253
Milner’s scheduler, 180, 373
Mobile parallel, 500–505
Mobility, 481–486, 497–506
Model checking, 6, 143, 161, 184 bounded, 186
Monotonicity and refinement, 39
is implied by distributivity, 233
Moore’s Law, 186, 377
Murray, Toby, 380
Mutual exclusion, 420, 439, 450, 452, 468, 471, 478

N
No junk principle, 251
NoEqT, 393–395, 397, 399, 406, 453, 454
Non-blocking ring, 82, 83, 341
Non-destructiveness, 234, 243
Non-interference, 127, 128, 172
Nondeterminism, 10, 11, 19, 192, 236, 250, 277
finite, 11, 277
inevitable in concurrency theory, 98
introduced by external choice, 10
introduced by hiding, 95, 97
introduced by renaming, 104
unbounded, 11, 99, 105, 225, 276, 278, 284
Nondeterministic selection ($), 394, 395
Norm (normality condition), 395
normal, 174, 175, 364, 369
Normal form, 161, 294, 305–315, 359
acceptances, 312, 313
failures, 313
refusal testing, 311, 313
revivals, 312, 313
traces, 361
Normalisation, 161, 175, 182, 357–363, 368, 378, 392, 415
lazy, 363
pathological, 161, 363

O
OCCAM, 83, 317
off argument, 203, 205, 207, 214, 216
Olderog, Ernst-Rüdiger, xi, 260
Ω (Omega): terminated process, 201
on argument, 203–205, 208, 214–216, 299, 301
Operational semantics, 41, 189, 191–228, 236, 247, 263, 287, 294, 297, 299, 392
congruence with denotational semantics, 252
Ouaknine, Joël, 323, 345
Output, 7–9, 15, 324, 326
Overseeer, 469–471, 475–480

P
Palikareva, Hristina, 186
Pantomime horse, 49, 56
Parallel
generalised ($\parallel$)
ex laws of, 303
operational semantics of, 206
Parallel operator
for adding constraints, 72
Parallel processes
can be sequentialised, 46, 51
Parameterisation, 55, 103
Parameterised verification problem, 385–415
Parameters, process, 20, 191
Parametricity, 392
Partial order, 229, 366
Pattern matching, 15, 20, 151
Peg solitaire, 67, 167, 186, 373
Peterson’s algorithm, 420, 450, 451
PHIL, 53
Phillips, Iain, 262
PHILS, 58
π-calculus, 241, 497–499
Piping ($\gg$), 110, 497
Index

Plotkin, Gordon, 200, 227
Pnueli, Amir, xi
Point-to-point communication, 94
Polymorphism, 65, 391
Port (mobile), 499, 500, 502
Prefixing, 3, 4
Prefix-choice, 7–9, 20
Pre-normal form, 358, 359, 378
Predicate symbol, 393, 399, 400
Prefix-choice, 7–9, 20
Predicates, 390
Predicates of, 390
Predicates and relations, 25
Predicates and relations, 25
prioritise, 489
Priority, 263, 340, 343, 483–497
Process algebra, 1, 26, 42, 194, 253, 293
Promotion of $\tau$ actions, 204
Protocol layering, 86, 457
Puhakka, Antti, 290
Quicksort, 112, 152, 185, 497
Railways, 18, 63
Rajamani, Sriram, 264
Recentness, specification of, 454
Recursion, 4, 5, 31, 100, 110, 232, 238
and fixed points, 31, 231–235
equational and $\mu$ styles, 5
guarded, see constructiveness
is not distributive, 25
laws of, 28
mutual, 5, 32, 34, 52, 234, 250, 251
infinite, 6, 57
operational semantics of, 203, 228
over $FP$, 257
parameterised, 8, 32, 47, 55
semantics of, 273
tail, see tail recursion
through sequential composition, 133
trace proof rules, 37
Reed, G.M. (Mike), 355
Refinement, 12, 43, 144, 185, 236, 242, 413,
462, 463, 465
failures, 117, 120
failures, as a proxy for $\equiv_{FP}$, 123
failures-divergences, 119–123
order, 233
properties of, 39

Q
Quicksort, 112, 152, 185, 497

R
Railways, 18, 63
Rajamani, Sriram, 264
Recentness, specification of, 454
Recursion, 4, 5, 31, 100, 110, 232, 238
and fixed points, 31, 231–235
equational and $\mu$ styles, 5
guarded, see constructiveness
is not distributive, 25
laws of, 28
mutual, 5, 32, 34, 52, 234, 250, 251
infinite, 6, 57
operational semantics of, 203, 228
over $FP$, 257
parameterised, 8, 32, 47, 55
semantics of, 273
tail, see tail recursion
through sequential composition, 133
trace proof rules, 37
Reed, G.M. (Mike), 355
Refinement, 12, 43, 144, 185, 236, 242, 413,
462, 463, 465
failures, 117, 120
failures, as a proxy for $\equiv_{FP}$, 123
failures-divergences, 119–123
order, 233
properties of, 39

Provable algebraically, 40
refusal testing, 270
revivals, 270
shared-variable, 457–463
trace, 39, 41, 120, 233
traces, 16, 457
Refinement check, simple, 146, 379
Refinement checking, 143, 161–163, 379
expressive power of, 379
Refusal set, 115, 116, 118, 161, 162, 199, 221,
237, 258, 262
maximal, 116, 358
may contain $\checkmark$, 138
Refusal testing (refusal traces) model
divergence-strict ($RT^{\downarrow}$), 274, 275, 290,
489
seeing beyond divergence ($RT^{\uparrow}$), 311
stable ($RT$), 262–266, 270, 272, 274, 282,
310, 312, 362, 489
refusals($P$), 115
Register transfer level, 496
Regular language, 42
Rehof, Jakob, 264
Relation, 105, 233
composition of, 105
creating models, 258
domain and range of, 105
Relations representing operators, 232, 233,
239, 240
Renaming ($\llbracket R \rrbracket$)
laws of, 300
Renaming ($\llbracket R \rrbracket$), 102–108, 112
$\checkmark$ never affected, 137
infinite-to-one, 226
injective, 103, 104
inverse function, 105
laws of, 103–106
non-injective, 104, 105
one to many, 106
operational semantics of, 206
relational, 105–108
traces of, 106
variable, 107
REPEAT, 7, 45, 48
Revival, 264, 361
Revivals model
divergence-strict ($R^{\downarrow}$), 282, 290
stable ($R$), 264–266, 268, 270, 271, 282,
296, 312, 358, 362
Rich trace model, 272, 274, 277
Right value, 427
Ring networks, 80
Roggenbach, Marcus, 43
Roundabout (traffic circle), 80
Rounds, Bill, 267
Routing, 73–86, 93, 98, 104, 340, 399
RUN, 16, 36, 40, 294, 321
Rushby, John, 480

S
Safety, see specification, safety
sat, 37, 39, 40
SAT checking, 70, 184, 186
SBDIA, 174, 364, 369, 371
Scattergood, Bryan, xii, 184, 191
Schneider, Steve, 321, 355
SCTree, 177, 182, 183
Security, ix, 127
Sequence notation, 30, 36
Sequential composition (;), 131–138
laws of, 132, 133, 300
operational semantics of, 206
traces of, 233
Sequential consistency, 454, 455, 457
Set as a parallel process, 158–160, 371
Shared variables, 419–480, 492
Σ (Sigma), 3, 18, 20, 94
Σ ✓, 132
Signal (undelayable) events, 138
Simpson, Hugo, 452
Simpson’s 4-slot algorithm, 451–457, 479, 480
SKIP, 16, 131, 193, 296
operational semantics of, 201, 207
Sliding window protocol, 90
Smolka, Scott, 362
Specification
behavioural, 17, 36, 38, 121, 281, 379
distributive, 380
failures, 117, 121, 240, 330, 343
failures-divergences, 121
in SVA, 438
liveness, 38
operators primarily for, 18
partial correctness, 38
processes as, 18
refinement closed, 380
safety, 38
total correctness, 38
trace, 36–40, 56, 57, 121
limitations of, 38
traces, 330
Stable failure, 237
Stable state, 115, 118, 175, 192, 193, 237, 241, 255, 259, 274, 358, 359
cannot perform ✓, 192, 221
Stack, 135, 381
State exploration, 6, 52, 210, 358
State explosion problem, 65, 158, 176, 319, 385–415, 483
State machine
finite, 6
State space, 8, 211
State-pair, 162
Statechart, 490–492
Stepwise refinement, 40
STOP, 3, 16, 27, 38, 40, 131, 196, 235, 238, 287
operational semantics of, 201, 207
Strong conflict, 76, 77, 265, 266
Strong order, 250, 273
Structural induction, 226
Structured compression, 182, 183
Structured operational semantics (SOS), 200–204, 208, 214
negative premises, 488
subs(a, e, P), 202, 208
Sudoku, 67–73, 107, 144, 169, 373, 481, 486, 490
application of renaming in, 113
Supercombinator, 161, 212–214, 215
Superstate, 400
SVA, 419–480
debugger, 436
front end, 434, 437
SVL, 434–436, 442, 443, 447, 450, 461, 468
Symmetry, 167, 168
Synchronisation tree, 194
Synchronous parallel (∥), 45–48
is not idempotent, 48
laws of, 45
traces of, 48

T
Tail recursion, 8, 34, 135
one step, 8
Tarski’s Theorem, 235, 273, 288
τ (tau) invisible action, 19, 41, 94, 95, 115, 132, 203, 228, 259, 296, 340, 358, 365, 368, 370, 488
τ-loop, 199
τ-loop elimination, 365–367
tau.loop.factor, 174, 365, 367
τ priority (over tock), 339
tau.priority, 343
τ promotion, 208, 215
TCOPY1, 324, 326
TCOPY2, 324
TCOPY3, 324, 326
TCOPY4, 324, 325
Telephone exchange (mobile CSP), 502
Temporal logic, 144, 257
Termination
as origin of revivals, 264
Termination (\(\triangleright\)), 131–138, 192–194, 206, 214, 228, 256, 260, 296
as a signal, 132, 138, 140–142, 170, 237, 346
in operational semantics, 204
simulation of, 217
under \(r\) priority, 340
Thread, 420, 424, 425, 428, 471
Threshold, 392–397
Throw (\(\Theta_A\)), 218, 290, 342, 463
laws of, 300, 140
operational semantics of, 207
Throw (\(\Theta_A\)), 139–141
Time out, 88, 96, 332, 352
Time: discrete vs continuous, 321–323
Timed CSP, 319, 321–323, 345–355
Timed systems, 321–355
Timing consistency check, 325, 330, 335, 343
Timing constraints, 324
inconsistent, 324, 326
tock, 488
tock-CSP, 321–340, 341–343
tock-CSP, metering, 341
TOCKS, 322, 325, 326
Token ring, 82, 83, 99, 179, 180, 183, 286, 341, 366, 368
Tools, 19–21, 64, 65, 112–114, 227
Trace
infinite, 255, 280
notation, see sequence notation
Trace models (\(T\)), 238
Traces, 16, 17, 29–40, 115, 321
denotational calculation of, 30
finite, 29
infinite, 29, 222–226, 272, 276
operational calculation of, 42
proof rules for, 37
Traces model
divergence-strict (\(T^\Box\)), 271, 282, 290
inadequacy of, 31, 33, 48, 115
infinitary divergence-strict (\(T^{\Box\omega}\)), 278, 280, 283
seeing beyond divergence (\(T^\#\)), 284, 287, 290
Traces (\(P\)), 223, 221
\(\text{traces}(P)\), 30–33, 36
\(\text{traces}_{\perp}(P)\), 222
Traces-plus model, 272, 274, 275, 278, 282, 285, 287, 312, 313, 357, 359, 414
Transitivity of refinement, 39
transparent, 175
Transputer, 185
Tree network, 75–80, 399, 408
Two-level timing model, 490
Two-phase automata, 494–497
U
UFP rule, see unique fixed points
Unbounded nondeterminism, see nondeterminism, unbounded
Ungranted request, 76, 78, 123, 124, 265
cycles of, 77
Uniform continuity, 380
Unique fixed points, 34, 46–48, 52, 59, 100, 101, 120, 133, 135, 232, 234, 250, 280
Uniqueness of models, 282
Unstable action, 260, 264
Unstable state, 116, 192, 255, 259, 296
Unstable visible event, 296
Urgent action, 324
Urgent state, 325
V
Valmari, Antti, 290
Van Glabbeek, Rob, 259, 267
Variable, 424
Vector, 32, 34, 35, 38
Virtual memory, 374
VLSI, 18
W
WAIT, 321
Walker, David, 421
Warnsdorff’s algorithm, 483–486
Watchdog, 166, 169, 184
WBUFF, 122, 400
Weak divergences strictness (WDS), 284
Wolf, goat and cabbage puzzle, 73
Wolper, Pierre, 415
Write when different, 449, 454, 457, 480
Y
Yantchev, Jay, 84
Z
Zero length buffer, 409