Index

Bold face page numbers are used to indicate pages with important information about the entry, e.g. the precise definition of the term or user-defined predicate, while page numbers in normal type indicate a textual reference.

!, 362–365
::, 32
$[X|Y]$, 188
[], 188
■, 144
□, 135, 141
$\Rightarrow$, 141
$\Rightarrow^*$, 313
$\equiv$, 21
$\lceil\ \rceil$, 118, 303, 343
$\leftrightarrow$, 14
$\lfloor\ \rfloor$, 101, 118, 303, 343
$\neq$, 102
$\wedge$, 13
$|\ |$, 60
`( -> ; )`, 230, 365
`=..`, **328**
%, 216
_, 12, 134, 188

accumulator, 197, 210, 329
Ackermann's function, 410
addkey, 194, 200
`alldifferent`, **110**, 190, 256, 268, 269, 271
`alldifferent_neq`, **190**, 254
anonymous variable (_), 188
answer, 141, **142**, 147
    generated, 400
    qualified, 316
answer redundant, 223, 225, 229
append, **189**, 317
arc consistency, **92**, 92–96
arc_consistent, **94**
arc_consistent_primitive, **94**, 110
arc_solv, **95**
arg, **328**
`arithmetic`, 299, **326**
arithmetic constraint solver
    Gauss-Jordan, 20
    incremental, 376
    incremental Gauss-Jordan, 353
    simplex, 71
arithmetic CSP, 98
artificial variable, 69
ask delay condition, 315, 428
`assert`, **332**
assignment problem, 266–269
association list, 193–198, 200, 204, 238, 274
    accessing a record, 193
    deleting a record, 194
    empty, 194
    inserting a record, 194
    modifying a record, 195
`atom`, **326**
attribute (of relation), 416

back_arc_solv, **96**
back_int_opt, **117**, 305
backtrack stack, 355
backtrack_solve, **90**
backtracking
    chronological, 89
    in evaluation, 147
    semantic, 360

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

backtracking goal solver, 350
  for cut, 364
  for minimization
    backtracking, 368
    retry, 366
  incremental, 355
backtracking integer optimizer, 305
base (in nucleotide), 31
basic feasible solution, **65**
basic feasible solved form, **65**, 377
basic variable, **65**
bb_int_opt, **119**
binary CSP, 88
binary search tree, 199, 200
  finding an item (`find`), 199
  inserting an item (`insert_bst`), 200
binary tree, 198–201, 203
  layout, 203–207
  search, *see* binary search tree
  traversal (`traverse`), 198
Bland's anti-cycling rule, 72
blocks world, 32–33, 61
body of rule, *see* rule
bool_solve, **30**, 35
Boolean
  constraint, 27–30
  solver, **30**, 317
bottom-up, **402**
bottom-up evaluation, 395–404
  mimicking top-down, 407–412
  termination, 412–416
  with equivalence, 401
  with subsumption testing, 414
bounds consistency, **98**, 97–109
bounds propagation, 97–109
bounds_consistent, **106**
bounds_consistent_addition, **100**
bounds_consistent_inequality, **101**
bounds_consistent_primitive, 110
branch and bound, 119, 301–305
building a house, **17**, 62, 195, 196
built-ins, 179, 230, 234, 303, 308, 309, 324–342
  database, 332–338, 341
  debugging, 338–342
  examining variables, 326–332
  input/output, 324–326
butterfly, *see* options trading

`c_add`, **186**, 294
`c_mult`, **186**, 294
`call`, **308**, 321
call option, *see* options trading
canonical form, **73**
canonical form simplifier, 77, 412
cardinality (| |), 60
Cartesian product, 416
CCP, 427
ceiling (⌈ ⌉), 118, 303, 343
children of a tree, 22
choice, 169–174
choicepoint, **143**, 358, 362
CHR, 430
chronological backtracking, 89
`clause`, **332**
CLP scheme, **151**, 425
CLP(FD), **152**, 251
CLP(Tree), **151**
CLP($\mathcal{R}$), **152**, 167
CNF, *see* conjunctive normal form
combining models, 270
comment (`%`), 216
complete solver, *see* solver
complex number, 186, 294
`compound`, **326**
`concat`, **294**, 295, 322
concatenation (::), 32, 189, 294
concurrent constraint programming, 427
conjunction (∧), **13**, 34
conjunctive normal form, 30, 79, 436
cons, 22, 188
consistency, 91, 92
  arc, **92**, 92–96
  bounds, **98**
  generalized, 109–114
  hyper-arc, **97**, 283

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

node, **92**, 92–96
constant, 22
constrain and generate, 253, 275
constrained lambda calculus, 434
constraint, **13**, 11–15
Boolean, 27–30
complex, 110, 256–258
disjunctive, 138
equivalent, **14**
primitive, **12**, 38
projection, **54**, 53–60
redundancy-free, **52**
redundant, **52**, 223, 276, 278, 431
answer redundant, 223, 225, 229
in rule, 223
solver redundant, 225, 226, 239
reified, 284
satisfaction, 18–21
satisfiable, **14**
sequence, 31–32, 294
solved form, 20
solver, *see* solver
store, *see* constraint store
tree, **24**, 22–27
unsatisfiable, **14**
user-defined, **134**, 133–137
deterministic, 221
constraint based graphics, 442
constraint database, 391, 393
constraint domain, 12
Real, 12
Tree, 22
blocks world, 32
Boolean, 27
sequences, 31
constraint handling rules, 430
constraint logic program, *see* program
constraint satisfaction problem, *see* CSP
constraint solver, *see* solver
constraint solving toolkits, 440
constraint store, **141**, 353, 355, 366
saving and restoring, 358–361

contig, 31
crypto-arithmetic problem, 254, 287
CSP, **86**, 86–88, 275
arc consistent, **92**
arithmetic, 98
binary, 88
bounds consistent, **98**
false domain, **92**
hyper-arc consistent, **97**
node consistent, **92**
valuation domain, **92**
`cumulative`, **112**, 256, 278
cut (!), 362–365

data structures, 185–207
association list, *see* association list
list, *see* list
record, *see* record
tree, *see* binary tree
database
constraint, 391, 393
relational, *see* relational database
spatial, 392
temporal, 393
`debug`, **339**
debugging, 338–342
definition of a predicate, 135
delay condition, 315, 430
ask, 315, 428
tell, 428
`delay_until`, **315**
delaying literal, **315**
`deleteff`, **260**
`deleteffc`, 261
`delkey`, 194
`deriv`, **299**
derivation, **141**
failed, **143**, 146
infinite, 146, 149
safe, **316**
successful, 316
selection, **154**
simplified, **150**

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

step, *see* derivation step
successful, **142**, 146
derivation step, **141**
if-then-else, **230**
minimization, **180**
negative, **309**
once, **234**
selection, **154**
derivation tree, **143**, 143–151, 213
simplified, **150**, 217
determined solved form, **37**
determined variable, **36**, 37
determines, **38**, 39
deterministic, 221, 222, 230, 318
difference list, 210
disequation ($\neq$), 102, 267
solver, 310, 315, 330
DNA mapping, 31
dom, 259, 261, **327**
domain, **86**, 252
false, **92**
for relational database, 416
range, **98**
valuation, **92**
domain splitting, 264, 306
don't care non-determinism, 429, 430, 438
don't know non-determinism, 426, 428
dump, 80, 325
dynamic scheduling, 157, 313–319, 330, 343
goal-based, 315
predicate-based, 315

electric circuit, 15, 202
RLC, 201, 202
element, **114**, 256, 268–271
eliminate, **354**, 370
empty goal ($\Box$), **135**, 141
■, 144
empty list ([]), 188
empty_alist, **194**, 200
emptylist, 428
enabled literal, 315
entry variable, 66
equiv, **77**
equivalence ($\leftrightarrow$), **14**, 75–78
fact, **399**
projected, **58**
tester, **77**, 412
eval, **354**
evaln, **299**
evaluation, 140–143, **147**, 349–352
algorithm, 350, 355
backtracking, 147
bottom-up, 395–404
bottom-up vs top-down, 405–406
incremental solver, 352–357
top-down, 140–143
exit variable, 66

fac, **139**, 141, 143, 144, 148, 150, 154, 160, 164
fact, **135**, 398
equivalence, **399**
head normal form, **398**
simplification rules, 400
simplifier, **399**
subsumed, **414**
fail, **308**, 335, 365
failed derivation, **143**
false, **13**, 34, 141, 151, 154
false domain, **92**
find, **199**
finite failure, **146**
finite tree, **23**
first fail principle, 261, 263, 281, 291
first phase problem, 69
fixed variable, 214
floor ($\lfloor\ \rfloor$), 101, 118, 303, 343
floundered, 316
Fourier elimination, 55, 56, 60
fourier_eliminate, **57**
fourier_simplify, **57**, 60, 63
free variable, 214
fsimpl, **399**
functional languages, 433

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

`functor`, **328**

Gauss-Jordan elimination, 20, 353
gauss_jordan, **21**, 27
gauss_jordan_simplify, 73, **74**, 78
gauss_jordan_solve, **21**, 35, 159
gen_ans, **400**
gen_facts, **402**
generate and test, 256
goal, **135**
  answer to, *see* answer
  empty (□), **135**, 141
  rewriting of, **136**
goal evaluation, *see* evaluation
`ground`, 315, **326**
guard, 428, 430

head normal form, **398**
head of rule, *see* rule
height of tree, **23**
hierarchical constraints, 439, 442
hierarchical modelling, 201–203
higher-order predicate, 307
hyper-arc consistency, **97**, 283

if-then-else, **230**, 362, 365
`iff`, 271
immediate consequences, 401
imperative languages, 439
impl, **76**
implication (→), 52, 75–78
  projected, **414**
  tester, **76**
inc_arith_solve, **382**
inc_bounds_solv, **375**
inc_gauss_jordan_solve, **354**
inc_simplex, **378**
inc_solve_goal, **356**
inc_tree_solve, **371**
incomplete solver, **35**
incremental arithmetic solver, 381
incremental Gauss-Jordan solver, 353
incremental solver, **353**
  arithmetic, 381
  bounds consistency, 374
  Gauss-Jordan, 353, 358, 360
  simplex, 377
  tree constraints, 371
independence of negative constraints, **416**
`indomain`, 259
infinite tree, **23**
`insert_bst`, **200**
`integer`, 303, **326**, 343
intended mode of usage, **215**
`intersect`, **333**, 339
`is_compatible`, **311**
isolv, **353**, 355
iteration, 174–179

Kaleidoscope, 439
Kirchhoff's Laws, 15

`labeling`, 253
labelling, 258–266, 271, 280
  domain splitting, 264
  value order, 259, 261
  variable order, 259, 261
`labellingff`, **260**
`length`, **261**, 262
`less_than`, 199, 200
library predicates, *see* built-ins
linear constraint, 19
linear expression, 19
linear inequality, 19
list, 188–193
  notation, 188
`list_append`, **196**
literal, **135**
  delaying, **315**
  enabled, 315
  if-then-else, **230**
  minimization, **179**
  negative, **309**
  once, **234**
literal ordering, 220
literal selection
  fair, 157
  independence, 153–157

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

right-to-left, 154
strategy, **316**
safe, **316**
local propagation, 35–41, 317
local_propagation_solve, **39**
`lookup`, 194, 200, 205, 277

map colouring problem, 86
matrix, 191
max_D, **98**
`maxdomain`, 259, 264, **327**
maximal bipartite matching, 112
`member`, **193**, 199, 259, 263, 312, 333
meta-program, 320
meta-programming, 320–323
metal plate problem, 185, 191, 444
min_D, **98**
`mindomain`, 259, 264, **327**
minimization, *see* optimization
`minimize`, **179**, 180, 204, 227, 276, 336, 366
mode of usage, **214**, 220, 222, 227, 232, 314
intended, **215**
satisfies, **214**
modelling, 15–18, 86–88, 167–180
a function, 173
a graph, 195
adjacency list, 195
choice, 169–174
combining models, 270
different models for a problem, 114, 266–272
comparing models, 269
electric circuit, 15, 202
hierarchical, 201–203
iteration, 174–179
list, *see* list
matrix, 191
negation, 309–313
optimization, 179–180
record, *see* record
relation, 252
symmetry, 289
vector, 294, 329
modelling language, 426
mathematical, 443–445
`modkey`, 195
monotonic solver, **35**, 156
`mortgage`, 175, 178

N-queens problem, 86, 262
natural join, 418
`ne`, 310, 314, 315, 330, 343
negation, 309–313
negative subgoal, **309**
`new`, **402**, **414**
Newton-Raphson method, 297, 300
nil, 22, 188
`nl`, **325**
node, 23, 198
node consistency, **92**, 92–96
node_arc_consistent, **95**
node_consistent, **94**
node_consistent_primitive, **94**, 110
`nodebug`, **339**
non-parameter, 20
`nonemptylist`, 428
`nonvar`, 315, 323, **326**
normalising solver, 20
`nospy`, **339**
`nospyall`, **339**
`not`, **309**, 364
`not_empty`, **294**, 322
`not_member`, **190**, 309
`notrace`, **339**
NP-complete, 29
null, 23, 198

objective function, **61**
occurs check, 373
Ohm's Law, 15
old-fashioned marriage problem, 87, 252, 283
`once`, **234**, 274, 335, 364
optimal solution, **61**
optimization, 61–72, 114–120, 179–180, 204, 227, 301–307, 445

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

implementation, 336, 366–369
integer, 114–120
backtracking, **117**, 368
branch and bound, **119**, 301–305
retry, **115**, 366
optimistic partitioning, 306
problem, **61**, 64, 179
simplex, **67**
options trading, 170–174, 210
butterfly, 172, 174, 180
call option, 170
put option, 170
straddle, 180
Oz, 434

parallel_resistors, **134**, 135, 136
parameter, 20, 65
partial solution, **54**, 134
pivoting, 66
predecessors, **196**, 234, 277
predicate, **134**
definition of, 135
higher-order, 307
preferred valuation, **61**
primitive constraint, **12**, 38
primitives, **15**, 34, 35
principle of least commitment, 264
probabilistic algorithm, 30
problem
NP-complete, 29
optimization, *see* optimization
satisfaction, **18**
solution, **18**
process view of CLP, 427
product, 418
program, **135**
program consequences, 401
projected equivalence, **58**
projected implication, **414**
projecting simplifier, **60**
projection, **54**, 53–60
for relational databases, 417
of constraint, **54**
propagation
bounds, 97–109
local, 35–41
propagation rules, 99–105
for minimum, 102
for addition, 99
for disequations, 102
for multiplication, 103, 104
put option, *see* options trading

qualified answer, 316
query
on database, 418
query transformation, 410

range, for finite domains, **98**
read, **324**
Real, 12, 152
real_opt, 118
record, 186–188, 202
recursion, 139, 175, 189, 206, 220
redundancy, **52**
redundancy-free simplifier, **60**
reified constraint, 284
relation, 169–170, 416
relational database, 170, 416–421
attribute, 416
domain, 416
natural join, 418
product, 418
projection, 417
query, 418
relation, 416
relational expression, 418
selection, 418
union, 417
relational expression, 418
renaming, **34**, 136, 141, 154
application of, 34, 136
retract, **333**
retry integer optimization, **115**
retry_int_opt, **115**, 366
rewrite rules, *see* term rewriting
rewriting of goal, *see* goal, 141, 154

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

river crossing problem, 168
rule, 133, 134, **135**, 137–140
  body of, **135**
  head of, **135**
  recursive, 139, 198
rule ordering, 220

same, **402**
SAT, 29
satisfaction problem, **18**
satisfiable, **14**
satisfiable, **38**, 39, 89, 92
scheduling, 251, 272–281, 431–433
search space, **146**
seesaw problem, 257, 265, 289
selected literal, **154**
selection, 418
selection derivation, 316
semantic backtracking, 360
sequence constraint, 31–32, 294
set based solver, **35**, 154
shadow of space, 55
simpl, **58**, 142, 149, 150
simple_solve_goal, **350**
simplex algorithm, **67**, 63–72, 376
  artificial variable, 69
  basic feasible solution, **65**
  basic feasible solved form, **65**
  basic variable, **65**
  Bland's rule, 72
  entry variable, 66
  exit variable, 66
  first phase problem, 69
  form of constraints, **64**
  incremental, 377
  pivoting, 66
  slack variable, 65
  solver, **71**, 377
simplex_opt, **68**
simplex_solve, **72**
simplification, 51–60
simplified derivation, **150**
simplified derivation tree, **150**, 217
simplified state, **150**
simplifier, **58**, 57–60, 151
  canonical form, **73**, 72–74, 77, 412
  properties, 60
    projecting, **60**, 152
    redundancy-free, **60**
    weakly projecting, **60**, 152
slack variable, 65
smm, **4**, **254**, 260
smuggler's knapsack problem, 101, 108, 114, 115, 117, 252, 304
solution, **14**
  basic feasible, 65
  of state or goal, **180**
  optimal, **61**
  partial, **54**, 134
  problem, **18**
solv, **34**, 141, 154
solved form, **20**
  basic feasible, **65**
  determined, 37
solver, 18, **34**, 33–35, 151
  arc consistency, 281
  backtracking, 89, 96
  Boolean, **30**, 317
  bounds consistency, 281, 374
    complete, 108
    incomplete, 107
  branch and bound, 267
  complete, **35**, 89, 96, 253
  finite domain, 267
    backtracking, 89
    bounds consistency, 107, 108
    node and arc consistency, 95, 96
  Gauss-Jordan, 21, 353
  incomplete, 30, **35**, 95, 225, 252, 381
  incremental, *see* incremental solver
  local propagation, **38**, 317
  node and arc consistency
    complete, 96
    incomplete, 95
  normalising, 20, 53

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022

properties, 33–35
  monotonic, **35**, 156
  set based, **35**, 154
  variable name independent, **35**, 152
simplex, **71**
tree constraints, 25, 371
well-behaved, **35**, 141, 152
solver redundant, 225, 226, 239
spatial database, 392
`spy`, **339**
spypoint, 339
state, **141**
  critical, **150**
  fail, **142**, 144
  success, **142**, 144
store, *see* constraint store
straddle, *see* options trading
subsumed
  fact, **414**
successful derivation, **142**, 146
symbolic algebra package, 445–446
syntactic identity (≡), 21
syntactic object, 136

tell delay condition, 428
temporal database, 393
term, **24**
term equation, **24**
term rewriting, 436
  augmented, 436
  constrained, 438
time stamping, 358
top-down evaluation, 397
`trace`, **339**
trailing, 358
`traverse`, **198**, 205
Tree, 22, 151, 152
tree, **22**
  binary, *see* binary tree
  children of, **22**
  constraint, 22, **24**
  constructor, **22**, 188
  finite, 23
  height, **23**
  infinite, 23
tree_simplify, 59, **59**, 60, 73, 151
tree_solve, **26**, 35, 151
true, **13**, 34
`true`, **232**

underscore (_), 12, 134, 188
unify, **26**, 27, 53, 59, 370
union, 417
unknown, 30, **34**, 252
unsatisfiable, **14**
user-defined constraint, *see* constraint

valuation, **14**, 33
  extension, **54**
  preferred, **61**
valuation domain, **92**
`var`, **326**
variable, 11
  anonymous, 188
  artificial, 69
  basic, 65
  Boolean, 267
  determined, **36**, 37
  fixed, 214
  free, 214
  non-parameter, 20
  parameter, 20, 65
  procedural, 177
  slack, 65
variable name independent solver, **35**, 152
variant, **34**, 35, 136
vars, **14**
vector, 294, 296

weakly projecting simplifier, **60**
well-behaved solver, **35**, 141, 152
`write`, **325**

Copyrighted Material

Downloaded from http://direct.mit.edu/books/book/chapter-pdf/176584/9780262279161_cao.pdf by KU Leuven University user on 09 September 2022