CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

Index

about, *see* `rdf:about`
ABox (assertion box), 176, 217–228, 233, 279, 407
  reasoning
    consistency check, 221, 222, 291, 385
    instance check, 221, 225, 294
    instance retrieval, 221
    realisation, 221
    satisfiability check, 221
  satisfiability of a set of ABoxes, 262
  self-realising, 279, 280, 289, 351, 352, 356, 361, 374, 385
  tableau algorithm with ABoxes, 291–299, 385–387
  tableau state viewed as ABox, 260–262, 264, 267, 277, 288, 289, 346, 351, 352, 356, 362, 374, 382–384
absolute URI, 67, 75
abstract domain, 232, 434
abstract role, 232, 411
acyclic
  RBox, 238–240
  TBox, 207, 209, 212–213, 223, 393–395
`addIneqs`, $\mathcal{ALCN}$ tableau function, 333
$\mathcal{AL}$-normalised concept expression, 247
$\mathcal{AL}$ language, 183, 186, 246
$\mathcal{ALC}$ language, 189
$\mathcal{ALCN}$ language, 196, 254, 267, 305
$\mathcal{ALCN}$ tableau state, 268
`All`, $\mathcal{ALCN}$ tableau constructor function, 305, 324
`all`, Haskell function, 335, 338, 342
`allRule`, $\mathcal{ALCN}$ tableau function, 324
all different, *see* `owl:AllDifferent`
all values from, *see* `owl:allValuesFrom`
alternative, *see* `rdf:Alt`
ancestor node, 286, 337, 359, 366, 378
`ancestors`, $\mathcal{ALCN}$ tableau function, 338
anchor, 30
anchor text, 30, 47
`And`, $\mathcal{ALCN}$ tableau constructor function, 305, 306, 321
and-visible concept, 321
`andRule`, $\mathcal{ALCN}$ tableau function, 321
anonymous resource, 71, 77, 85, 92
`any`, Haskell function, 319, 329, 338
`apply`, $\mathcal{ALCN}$ tableau function, 317
`applyAll`, $\mathcal{ALCN}$ tableau function, 322
`applyAnd`, $\mathcal{ALCN}$ tableau function, 321
`applyAtLeast`, $\mathcal{ALCN}$ tableau function, 336
`applyAtMost`, $\mathcal{ALCN}$ tableau function, 330
`applyExists`, $\mathcal{ALCN}$ tableau function, 334
`applyOr`, $\mathcal{ALCN}$ tableau function, 331
arity, 72, 84
assertion, 176
  concept, 218, 260, 278, 279
  role, 218, 260, 279
    negated, 240
assertion box, *see* ABox
assertion property, *see* `owl:assertionProperty`
associative operation, 184, 187, 247, 306, 309
at-least rule, 266, 269, 287, 335, 370, 380
`AtLeast`, $\mathcal{ALCN}$ tableau constructor function, 305, 318, 336
at-most rule, 263, 266, 269, 326, 370, 374, 380, 386
`AtMost`, $\mathcal{ALCN}$ tableau constructor function, 305, 318, 330
`Atomic`, $\mathcal{ALCN}$ tableau constructor function, 305
atomic
  concept, 175, 179, 183, 185, 186, 203, 207, 210, 212, 221, 228
  negation, 183, 186, 211, 255
  role, 175, 182, 228
attribute, 18, 56, 58
attributive language, 183, *see also* $\mathcal{AL}$ language
axiom

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

class
disjointness, 427
equivalence, 427, 433
inclusion, 427
concept, 181, 183, 187
concept equivalence, 179, 203, 205
concept inclusion, 181, 203, 205, 206, 213, 392, 394, 430, 434, 437
definitional, 206, 212, 223, 235, 431
general concept inclusion, 206, 392, 430
property
equivalence, 437
inclusion, 435
role, 181, 183, 202, 206
role equivalence, 182, 183, 199, 203, 205, 438
role inclusion, 182, 183, 199, 203, 235, 239, 448
role transitivity, 205
terminological, 176, 179, 212–217, 281, 392
`Axiom`, $\mathcal{ALCN}$ tableau datatype, 311
axiom transformation, 392
`axiomConcept`, $\mathcal{ALCN}$ tableau function, 311

background knowledge, 47, 181, 206, 207, 235
backtracking, 143, 146, 347, 387
dependence directed, 398, 399
bag, *see* `rdf:Bag`
base algorithm (tableau), 348
base interpretation, 207
base symbol, 207, 209
base URI, 67, 78, 81
blank resource, 71, 77, 85, 92
blocked node, 285, 289, 359, 378, 400
directly blocked, 359, 378
indirectly blocked, 359, 360, 378, 386
blocking, 288, 335, 337–338, 350, 351, 361
base condition
for equality blocking, 359
for pairwise blocking, 365, 378
dynamic, 285, 359
equality, 359
pairwise, 365–368
static, 285, 286
subset, 285, 337, 351, 400
Boolean constraint propagation, 397
bottom, 180, 183, 186, 203, 204, 247, 258, 401, 413
branching factor, 276, 288, 381, 387
browser, 9

caching, 18, 400
cardinality, *see* `owl:cardinality`, `owl:maxCardinality`, `owl:minCardinality`, `owl:minQualifiedCardinality`, `owl:maxQualifiedCardinality`
cardinality constraint, 419–422
cardinality restriction, 422
global, 438
qualified, 447
catalogue, 40, 44–46, 110, 112
CGI (Common Gateway Interface), 14–21, 39
specification, 14
`check`, $\mathcal{ALCN}$ tableau function, 318
`childId`, $\mathcal{ALCN}$ tableau selector function, 307
choice point, 271, 314, 347, 375, 381, 387–399
clash, 258, 266, 268, 298, 318, 369, 374, 378, 387–401
`clash`, $\mathcal{ALCN}$ tableau function, 318
clash-free tableau state, 258, 292, 318
class, 86, 87, 92, 94, 103, 175, 408, *see also* `rdfs:Class`, `owl:Class`
complement, 425–426
disjointness, 161–162, 427, 433–434, 442
equivalence, 427, 433
Haskell type, 305, 339
hierarchy, 103, 158, 159, 181
inclusion, 427
intersection, 422–424
union, 424–425
classification (TBox reasoning), 177, 200, 211, 401
client, 9
cloaking, 40
closed world assumption, 223–227
$ClosH_T$, hierarchy closure, 356, 385
$ClosHT_T$, hierarchy and transitivity closure, 356, 362, 374, 385
ClosI, inverse closure, 361, 374, 385
$ClosT_T$, transitivity closure, 352, 356, 385
closure property of query language, 136, 137, 142
coherence (TBox reasoning), 210, 403
collection, 94, 414
`Colouring`, $\mathcal{ALCN}$ tableau type, 340
`colourings`, $\mathcal{ALCN}$ tableau function, 340
`colourNode`, $\mathcal{ALCN}$ tableau function, 343
colour partition, 326, 329, 340–342
comment, *see* `rdfs:comment`
Common Gateway Interface, *see* CGI
commutative operation, 187, 247
complement, 180, 187, 189, 298, 425–426
complement of, *see* `owl:complementOf`
complete tableau, 257, 270–272, 280, 289, 346, 381
completeness of tableau algorithm, 277–278, 288, 297, 336, 382–383
component, 197
composite concept, 175, 183, 185, 218, 390, 395
concatenation of lists, Haskell, 317
concept, 175, 179, 413
and-visible, 321
assertion, 218, 260, 278, 279
as tableau node label, 256
atomic, 175, 179, 183, 185, 207, 210, 212, 221, 228
composite, 175, 183, 185, 218, 390, 395

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

constructor, 179, 180, 183, 195, 203, 236, 280, 304, 390
definition of, 206, 212, 223, 431
internalisation concept of TBox, 213, 311
lazy unfolding of, 395
root, 256, 259, 267, 269, 347, 379
unfolding of, 256, 258, 320, 331, 390, 392
universal, 179, 183, 186, 203, 204, 305, 413
`Concept`, $\mathcal{ALCN}$ tableau datatype, 305
concept axiom, 181, 183, 187, 206, 212, 223, 431
elimination, 212, 213, 223
primitive, 392
concept disjointness (TBox reasoning), 176, 210
concept subsumption (TBox reasoning), 250
concept equivalence (TBox reasoning), 210
concept equivalence axiom, 179, 203, 205
concept expression, 176, 183
concept hierarchy, 176, 211, 401
concept inclusion axiom, 181, 203, 205, 206, 213, 392, 394, 430, 434, 437
concept intersection, 179, 183, 186, 203, 204, 256, 269, 320, 361, 379, 389, 391
concept name, 183, 218, 228
concept satisfiability (TBox reasoning), 176, 210, 212, 215, 222, 249, 254, 268, 272, 281, 287, 302, 312, 313, 346, 379, 385
concept subsumption (TBox reasoning), 176, 210, 250, 314, 401
concept union, 180, 189, 203, 204, 258, 269, 331, 361, 379, 389, 391, 392, 397, 398
`ConceptName`, $\mathcal{ALCN}$ tableau type, 305
`concepts`, $\mathcal{ALCN}$ tableau selector function, 307
concept satisfiability (TBox reasoning), 249
concrete domain, 231–233, 411, 434, 443
concrete role, 232, 411
conjunction, 143, 155, 228
consistency check
in ABox reasoning, 221, 222, 291, 385
in TBox reasoning, 210, 212
constraint, *see also* OWL, constraint
propagation, 397
value constraint (SPARQL), 152
constructor
concept, 179, 180, 183, 195, 203, 236, 280, 304, 390
role, 182, 199, 234
constructor function, Haskell, 303, 305, 316
contradiction (in a tableau), *see* clash
`Contributor`, Dublin Core, 111
copy-in model, 365–368, 371–373, 384
`Coverage`, Dublin Core, 111
crawler, 22, 39, 40, 43, 125
trap, 39, 44
`Creator`, Dublin Core, 110
CSS (cascading style sheet), 62
cyclic
model, 364
RBox, 238
TBox, 207, 281

DAML ontology language, 407
DAML+OIL ontology language, 108, 109, 407
dangling page, 34
`data`, Haskell keyword, 305
data range, *see* `owl:DataRange`
database, 38, 44, 131, 136, 223–227
datatype
Haskell, *see* Haskell datatype
OWL, *see* OWL, datatype
RDF, *see* `rdf:datatype`
datatype property, *see* `owl:DatatypeProperty`
`Date`, Dublin Core, 111
decidable logic, 178, 183, 202, 237, 238, 446
declarative language, 127, 135
deep web, 37–39, 41, 44, 54
defined, *see* `rdfs:isDefinedBy`
definition, 186
concept, 206, 212, 223, 431
role, 235
definitional axiom, *see* axiom, definitional
definitional part of TBox, 207, 394
definitorial TBox, 207–209, 395
dependence directed backtracking, 398, 399
descendant node, 286, 378, 400
description, *see* `rdf:Description`
`Description`, Dublin Core, 111
Description Logic languages, 175
$\mathcal{AL}$, 183, 186, 246
$\mathcal{ALC}$, 189
$\mathcal{ALCN}$, 196, 254, 267, 305
$\mathcal{RIQ}$, 238
$\mathcal{SHIQ}$, 197, 203, 227, 375
$\mathcal{SROIQ}$, 239
deterministic rule, 270, *see also* non-deterministic rule
different, *see* `owl:AllDifferent`
different from, *see* `owl:differentFrom`
directly blocked node, 359, 378
directory, 22, 45, 112
disjointness, 161–162, 176, 210, 427, 433–434, 442, 448
of concepts (TBox reasoning), 176, 210
disjoint roles, 240
disjoint with, *see* `owl:disjointWith`
disjunction, 155, 189, 397, 399
distinct members, *see* `owl:distinctMembers`
distinct nodes, 266, 318, 335, 369, 374, 379
document, 9
domain
abstract, 232, 434
concrete, 231–233, 411, 434, 443
of interpretation, 185, 186, 203, 219

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

RDF, *see* rdfs:domain
Dublin Core, 110–111, 113, 115, 130
dynamic blocking, 285, 359
dynamic link, 38, 39, 44
dynamic page, 17, 38

edge
RDF graph, 71, 75
tableau graph, 255, 268, 307, 378
Edge, $\mathcal{ALCN}$ tableau datatype, 307
edges, $\mathcal{ALCN}$ tableau selector function, 307
elem, Haskell function, 319
embedding RDF in HTML, 123–125
entity (XML), 60
enumeration class, 443
equality blocking, 359
base condition, 359
equality of individuals, 219, 227
equivalence
class, 427, 433
concept, 179, 203, 205
(TBox reasoning), 210
property, 437
role, 182, 183, 199, 203, 205, 438
equivalent axioms, 188, 189
equivalent classes, *see* owl:equivalentClass
equivalent languages, 193–195
equivalent property, *see* owl:equivalentProperty
existential restriction
full, 190, 203, 204, 418
limited, 179, 183, 186
rule, 257, 269, 287, 332, 361, 370, 379
Exists, $\mathcal{ALCN}$ tableau constructor function, 305, 334
expansion
of TBox, 208, 212
query, 46–47
expansion rule, 270, 286, 292, 317, 359, 360

FastCGI, 20
filter, Haskell function, 323
findNode, $\mathcal{ALCN}$ tableau function, 323
finite-model property, 203, 362, 363
finite-tree model property, 362
firing condition of tableau rule, 270, 314, 379
first, *see* rdf:first
first-order logic (FOL), 155, 177, 185, 227–231
foldl, Haskell function, 325, 335, 336
foldl1, Haskell function, 312, 325
Format, Dublin Core, 111
formatting rule, 62
forms, 17
full existential restriction, 190, 203, 204, 418
full negation, 180, 190, 203, 204, 211, 250, 375, 395, 425, 426
full-text search, 26, 46

functional property, *see* owl:FunctionalProperty
functional restriction, 202

general concept inclusion (GCI) axiom, 206, 392, 430
general role inclusion axiom (g-RIA), 235, 239, 448
global property restriction, 443
graph
ABox, 224
class hierarchy, 103
colouring, 326–331, 340–342
concept hierarchy, 176, 401–403
RDF, 68, 70, 141
edge, 71, 75
label, 71, 107, 141
node, 71
RDF graph pattern, 141–143, 151
role hierarchy, 198
tableau, 224, 255, 268, 306, 346, 378

has self, *see* owl:hasSelf
has value, *see* owl:hasValue
Haskell, functional language, 302–345
constructor function, 303, 305, 316
datatype, 303–305
function
all, 335, 338, 342
any, 319, 329, 338
elem, 319
filter, 323
foldl, 325, 335, 336
foldl1, 312, 325
map, 312
minimum, 329
sort, 339
tails, 319
infix function, 319
keyword
data, 305
infixl, 306
where, 312, 341
lazy evaluation, 314
list, 312, 315
list comprehension, 319
list concatenation, 317
operator, 306
++, 317
., 340
||, 319
record manipulation syntax, 321
type class, 305, 339
type declaration, 308
type synonym, 306
type variable, 339
head of Haskell list, 315
heir of tableau node, 372, 384
heuristic

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

MOMS, 399
oldest-first, 399
hierarchy
class, 103, 159, 181
concept, 176, 211
property, 103, 159
role, 198, 200, 202, 214, 223, 352–356
regular, 241
XML, 56
higher-order RDF statements, 87
homepage, 9
homonym, 27
HTML, 9, 10, 12, 15, 17, 21
attribute, 18
specification, 10, 53
HTTP, 11, 15
specification, 15
hyperlink, 10
hypertext, 9, 11, 21

`Id`, $\mathcal{ALCN}$ tableau type, 307
ID, *see* `rdf:ID`
IdBlock$_{\mathbf{T}}$, 351, 352, 355, 362
identifiable nodes, 263, 266, 268, 326, 327, 329, 369, 374, 378, 386
identifier, *see* uniform resource identifier (URI)
`Identifier`, Dublin Core, 111
`identifyNodeList`, $\mathcal{ALCN}$ tableau function, 329
`identifyNodes`, $\mathcal{ALCN}$ tableau function, 328
implicit disjointness, 162
`Implies`, $\mathcal{ALCN}$ tableau constructor function, 311
inclusion
class, 427
concept, 181, 203, 205, 206, 213, 392, 394, 430, 434, 437
property, 435
role, 182, 183, 199, 203, 235, 239, 448
inconsistency, *see* consistency check
indexing, 24, 28, 44
stop word list, 25
indirectly blocked node, 359, 360, 378, 386
individual, 175, 186, 218, 219
individual name, 218, 219, 228, 260, 291, 346
`ineqs`, $\mathcal{ALCN}$ tableau selector function, 307
inequality of individuals, 219, 227
inequalities, tableau component, 266, 268, 292, 306, 346, 374, 378, 385, 388
infix function, Haskell, 319
`infixl`, Haskell keyword, 306
`initTableau`, $\mathcal{ALCN}$ tableau function, 312
instance, 86, 441–443
instance reasoning
ABox
instance check, 221, 225, 294
instance retrieval, 221
realisation, 221
RDF, 159–161
integer, *see* `rdf:Integer`
intensional, 430, 437, 441
intermediate resource, 71, 77, 85, 92
internalisation, 213–217, 223, 281, 287, 352, 385, 392, 395
internalisation concept, 213, 311
interpretation, 184–189, 203, 218, 255, 259, 413
base, 207
first-order, 230
natural, 279, 290, 352, 356, 362
intersection, 105, 176, 179, 183, 186, 203, 204, 234, 389, 391, 422–424
intersection of, *see* `owl:intersectionOf`
intersection rule, 256, 269, 320, 361, 379
inverse functional property, 439, *see also* `owl:InverseFunctionalProperty`
inverse of, *see* `owl:inverseOf`
inverse property, 438
inverse role, 182, 199–201, 203, 204, 356–362, 438
irreflexive role, 240
is defined by, *see* `rdfs:isDefinedBy`
iterative variant of tableau algorithm, 272, 381

knowledge base, 176
knowledge representation, 21, 69, 98, 176, 408

label, *see also* `rdfs:label`
RDF graph, 71, 107, 141
tableau graph, 255, 260, 268, 307, 369, 377, 378
`Language`, Dublin Core, 111
lazy evaluation, Haskell, 314
lazy unfolding of concepts, 395
lexical normal form, 319, 345, 390–392
limited existential restriction, 179, 183, 186
link, 10, 126
dynamic, 38, 39, 44
linking RDF to HTML, 125–127
list
Haskell, 307, 312, 315
list comprehension, 319
RDF, *see* `rdf:List`
list item, *see* `rdf:li`
literal, 56, 66, 68, 77, 141, 443
local property restriction, 445
logic
Description Logic, 175, 183
first-order logic, 177, 227–231
logic programming, 144
monotonic logic, 410

map
DL to first-order logic, 227–231
role-value, 236
tableau to ABox, 260–262, 264, 267, 277, 288, 289, 346, 351, 352, 356, 362, 374, 382–384

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

`map`, Haskell function, 312
markup language, 9, 21, 55–65, 407
max cardinality, *see* `owl:maxCardinality`
max qualified cardinality, *see* `owl:maxQualifiedCardinality`
meaning, *see* semantics
meaning of concepts and roles, *see* interpretation
meta-information, 47, *see also* metadata
metadata, 52–54, 66, 110, 114, 123, 127, 158
metalanguage, 55–65
meta search engine, 43, 47
min cardinality, *see* `owl:minCardinality`
min qualified cardinality, *see* `owl:minQualifiedCardinality`
minimality principle
  in structural subsumption algorithm, 250, 253
  in tableau algorithm, 257, 261, 270, 290
`minimum`, Haskell function, 329
model, 188, 204, 219, 313, *see also* interpretation
  RDF data model, 69
model-based query language, 141–158
model construction, tableau algorithm, 279, 350, 355, 360, 371, 383
`models`, $\mathcal{ALCN}$ tableau function, 313
MOMS heuristic, 399
monotonic logic, 410
multiplicity, 201
MusicBrainz, 114

`_n`, *see* `rdf:_n`
name
  concept, 183, 218, 228
  individual, 218, 219, 228, 260, 291, 346
  qualified, 59, 74, 75, 78
  role, 182, 218, 228
  universal, 59
name symbol, 207, 209, 212, 395
namespace, 59, 74, 100
naming convention for roles, 182
natural interpretation, 279, 290, 352, 356, 362
negation
  concept
    atomic, 183, 186, 211, 255
    full, 180, 190, 203, 204, 211, 250, 375, 395, 425, 426
  normal form, 255, 267, 310, 376
  RDFLan, 155
  role, 182, 222, 234
negative property assertion, *see* `owl:NegativePropertyAssertion`
neighbour node, 358, 360, 369, 370, 378
`newEdge`, $\mathcal{ALCN}$ tableau function, 333
`nextId`, $\mathcal{ALCN}$ tableau selector function, 307
`nid`, $\mathcal{ALCN}$ tableau selector function, 307
nil, *see* `rdf:nil`
`nnf`, $\mathcal{ALCN}$ tableau function, 310
NNF, *see* negation normal form
node
  RDF graph, 71
  tableau graph, 255, 268, 307, 378
`Node`, $\mathcal{ALCN}$ tableau datatype, 307
node id, *see* `rdf:nodeID`
`nodes`, $\mathcal{ALCN}$ tableau selector function, 307
`noEdge`, $\mathcal{ALCN}$ tableau function, 342
nominal, 233–234, 412, 415
non-determinism, 347, 387–401
  don't care, 271
  don't know, 271
non-deterministic rule, 258, 261, 263, 266, 270, 292, 315, 317, 326, 331, 375, 381, 387
non-empty TBox, 281, 348
non-identifiable nodes, 266, 318, 335, 369, 374, 379
non-logical symbol, 228
`nonBlocked`, $\mathcal{ALCN}$ tableau function, 338
normal form
  for structural subsumption algorithm, 246
  lexical, 319, 345, 390–392
  negation, 255, 267, 310, 376
`Not`, $\mathcal{ALCN}$ tableau constructor function, 305
nothing, *see* `owl:Nothing`
number restriction, 202, 419
  qualified, 196, 201, 374, 380, 447
    at-least rule, 370, 380
    at-most rule, 370, 374, 380, 386
    choice rule, 375, 380
  unqualified, 195, 265
    at-least rule, 266, 269, 287, 335
    at-most rule, 263, 266, 269, 326

object, 70, 76, 87, 155, 234, *see also* `rdf:object`
object-oriented, 107–109
object-oriented modelling, 101, 201, 421
object property, *see* `owl:ObjectProperty`
OIL ontology language, 407
oldest-first heuristic, 399
one of, *see* `owl:oneOf`
ontology
  heavyweight, 66, 175, 408
  lightweight, 66, 98, 181, 408
ontology editor, 84, 442
ontology language
  DAML, 407
  DAML+OIL, 108, 109, 407
  OIL, 407
  OWL, 407–449
on class, *see* `owl:onClass`
on property, *see* `owl:onProperty`
Open Directory Project, 45, 112
open world assumption, 223–227, 410
optimisation, 163–168, 387–404
`Or`, $\mathcal{ALCN}$ tableau constructor function, 305, 306, 331
outbound hyperlink, 31, 36
`owl:`
  `AllDifferent`, 441

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

allValuesFrom, 416
assertionProperty, 448
cardinality, 421
Class, 408, 413
complementOf, 409, 425
DataRange, 415, 443
DatatypeProperty, 434
differentFrom, 441
disjointWith, 427, 433, 449
distinctMembers, 442
equivalentClass, 427, 430
equivalentProperty, 437
FunctionalProperty, 438
hasSelf, 447
hasValue, 418
intersectionOf, 422, 436
InverseFunctionalProperty, 439
inverseOf, 438
maxCardinality, 419
maxQualifiedCardinality, 447
minCardinality, 420
minQualifiedCardinality, 447
NegativePropertyAssertion, 448
Nothing, 413
ObjectProperty, 434
onClass, 447
oneOf, 414, 415, 419
onProperty, 415
propertyChainAxiom, 448
propertyDisjointWith, 448
qualifiedCardinality, 447
Restriction, 415
sameAs, 441
someValuesFrom, 417
sourceIndividual, 448
SymmetricProperty, 440
targetIndividual, 448
Thing, 413, 441
topObjectProperty, 448
TransitiveProperty, 441
unionOf, 424
OWL (Web Ontology Language), 66, 106, 108, 109, 158, 175, 407–449
axioms, 427
class description, 413
complement, 425–426
enumeration, 414–415
intersection, 422–424
named class, 413
property restriction, 415–422
union, 424–425
constraint
cardinality, 419–422
value, 415–419
datatype, 443
enumeration class, 414
OWL 1 specification, 408
OWL 2 specification, 408
property
datatype, 410, 434
object, 410, 434
sublanguages, 411
OWL DL, 412
OWL Full, 411
OWL Lite, 412

PageRank algorithm, 30–36, 97
pair, $\mathcal{ALCN}$ tableau function, 329
pairwise blocking, 365–368, 378
base condition, 365, 378
parameters of a tableau rule, 270, 381
parentId, $\mathcal{ALCN}$ tableau selector function, 307
parse type, *see* rdf:parseType
PartialColouring, $\mathcal{ALCN}$ tableau type, 341
partial application of Haskell functions, 309
Partition, $\mathcal{ALCN}$ tableau type, 340
precision, 27
predecessor node, 286, 378
predicate
first-order, 228
RDF, 70, 87, 155
prefix
rdf:, 74
XML namespace, 59
preprocessing ABox tableaux, 292
preserving satisfiability of tableaux, 261, 278, 347, 362
primitive concept axiom, 392
projection, 139
Prolog language, 127, 144, 155
property, 69, 103, 175, *see also* rdf:Property and rdfs:Property
characteristics, 440–441
disjointness, 448
domain, *see* rdfs:domain
equivalence axiom, 437
hierarchy, 103, 159
inclusion axiom, 435
range, *see* rdfs:range
restriction, 104–106, 415–422
global, 106, 443
local, 445
universal, 448
property chain axiom, *see* owl:propertyChainAxiom
property disjoint with, *see* owl:propertyDisjointWith
Protégé ontology editor, 84
Publisher, Dublin Core, 111

qualified cardinality, *see* owl:qualifiedCardinality
qualified name, 59, 74, 75, 78
qualified number restriction, 196, 201, 203, 204, 374, 380, 447

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

query, 142, 223
  expansion, 46–47
  language, 122–168
    database-oriented, 136–141
    model-based, 141–158
    XML, 130, 135
  optimisation, 163–168
  processing, 25–28

Racer, 429
range, *see* `rdfs:range`; `owl:DataRange`
ranking, 22, 27–36
  PageRank algorithm, 30–36, 97
RBox (role box), 181
  acyclic, 238–240
  cyclic, 238
RDF (Resource Description Framework), 65–97, 110–118, 175, 408
  collection, 94, 414
  complex query, 142
  container, 90–95, 137–141, 150
  data model, 69
  description, 70
  domain, 160, 161
  embedding in HTML, 123–125
  global property restriction, 106
  graph, 68, 70
    edge, 71, 75
    label, 71, 107, 141
    node, 71
  graph pattern, 141–143, 151
  higher-order statements, 87
  instance, 86
  linearisation, 68
  linking to HTML, 125–127
  non-binary relations, 84
  Notation 3 syntax, 84
  object, 70, 76, 87, 155
  predicate, 70, 87, 155
  primitive query, 142
  query languages, 122–168
  range, 160, 161
  reification, 87–90, 149, 155
  resource, 66, 69, 71, 72, 75, 77, 79, 84–87, 91, 92, 98, 100, 103, 107, 411
  schema, 66, 86, 98–110, 141, 158, 159, 181, 407, 408
    specification, 98
  serialisation, 68, 72, 84
  specification, 66
  subject, 70, 87, 155
    shared, 75
  tag, *see* `rdf:RDF`
  triple, 70
  type, 95
  typed literal, 96
  XML syntax, 73–84
`rdf:`
  `about`, 74
  `Alt`, 91
  `Bag`, 91, 124
  `datatype`, 96, 106, 136, 232, 419, 443
  `Description`, 74, 76
  `first`, 94
  `ID`, 79–84, 90
  `Integer`, 140
  `li`, 92
  `List`, 94, 443
  `_n`, 91
  `nil`, 94
  `nodeID`, 77
  `object`, 88
  `parseType`, 77, 95, 410, 414
  `predicate`, 88
  `Property`, 103, 434, 439
  `RDF`, 73, 74
  `resource`, 76
  `rest`, 94
  `Seq`, 91
  `Statement`, 87
  `subject`, 88
  `type`, 86, 91, 100
  `value`, 85
RDFizer, 97
RDFLan, 155–156, 162, 163
`rdfs:`
  `Class`, 100, 408
  `comment`, 107
  `domain`, 104–106, 158, 436, 443
  `isDefinedBy`, 107
  `label`, 107
  `range`, 104–106, 158, 436, 443
  `Resource`, 105, 413, 441
  `seeAlso`, 107
  `subClassOf`, 101, 106, 159, 427–430
  `subPropertyOf`, 103, 106, 159, 435, 437
RDFS, *see* RDF schema
RDQL (RDF Data Query Language), 147–150
realisation (ABox reasoning), 221
reasoning, 42, 53, 98, 156–162, 176, 209–217, 220–223, 246–299, 302–404
  ABox
    consistency check, 221, 222, 291, 385
    instance check, 221, 225, 294
    instance retrieval, 221
    realisation, 221
    satisfiability check, 221
  TBox
    classification, 177, 200, 211, 401
    coherence, 210, 403
    concept disjointness, 176, 210
    concept equivalence, 210

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

concept satisfiability, 176, 210, 212, 215, 222, 249, 254, 268, 272, 281, 287, 302, 312, 313, 346, 379, 385
concept subsumption, 176, 210, 250, 314, 401
consistency check, 210, 212
recall, 27
record manipulation syntax (Haskell), 321
recursive variant of tableau algorithm, 271
redirection-based model, 290, 351, 358, 364
reflexive role, 239
reflexive–transitive closure, 354, 360, 385
regular expression, 45, 151, 155
regular role hierarchy, 241
`Relation`, Dublin Core, 111
relation, 72
binary, 70, 175, 185, 227
non-binary, 84
unary, 175, 227
relative URI, 67, 78, 81
`removeNode`, $\mathcal{ALCN}$ tableau function, 323
resource, 11, 53, *see also* RDF, resource
Resource Description Framework, *see* RDF
rest, *see* `rdf:rest`
restriction
cardinality, 422, 438, 447
existential, 179, 183, 186, 190, 203, 204, 418
functional, 202
global property, 443
limited existential, 179, 183, 186
local property, 445
number, 195, 196, 201, 202, 265, 374, 380, 419, 447
property, 104–106, 415–422, 443, 445
qualified number, 196, 201, 203, 204, 374, 380, 447
role, 234, 236
self, 240
universal, 180, 416
unqualified number, 195, 265
value, 180, 183, 186, 203, 204, 417
rewrite rule, 45
`Rights`, Dublin Core, 111
$\mathcal{RIQ}$ language, 238
robot, 22–24
Robot Exclusion Standard, 23
role, 175, 179
abstract, 232, 411
assertion, 218, 260, 279
negated, 240
as tableau edge label, 256, 369, 377
atomic, 175, 182, 228
axiom, 181, 183, 202, 206
box, 181, 238, 240
composition, 234, 237
concrete, 232, 411
constructor, 182, 199, 234
definition, 235
depth, 276
disjointness, 240
equivalence axiom, 182, 183, 199, 203, 205, 438
expression, 182, 183
hierarchy, 198, 200, 202, 214, 223, 352–356
regular, 241
inclusion axiom, 182, 183, 199, 203, 235, 239, 448
intersection, 234
inverse, 182, 199, 438
irreflexive, 240
name, 182, 218, 228
naming convention, 182
negation, 182, 216, 222, 234
reflexive, 239
reflexive–transitive closure, 234, 235
restriction, 234, 236
simple, 202, 241, 441
symmetric, 200, 206, 240, 440
transitive, 197, 198, 202, 203, 205, 223, 240, 348–352
transitive closure, 197, 198, 234, 235
transitivity axiom, 205
union, 234
universal, 214, 234, 240, 448
`role`, $\mathcal{ALCN}$ tableau selector function, 307
role–value map, 236
`RoleName`, $\mathcal{ALCN}$ tableau type, 305
roles (function), 377
root concept, 256, 259, 267, 269, 346, 347, 379
root node
$\mathcal{AL}$-satisfiability, 249
tableau, 256, 269, 379
RQS (RDF Query Specification), 136–141
RSS, 115
rule, *see also* tableau, rule
$\bowtie$-, 375, 380
$\exists$-, 257, 269, 287, 332, 361, 370, 379
$\forall$-, 269, 322, 355, 361, 379
$\forall_+$-, 257, 349, 350, 355, 361, 379
$\geqslant$-, 266, 269, 287, 335, 370, 380
$\leqslant$-, 263, 266, 269, 326, 370, 374, 380, 386
$\sqcap$-, 256, 269, 320, 361, 379
$\sqcup$-, 258, 269, 331, 361, 379
formatting, 62
RDFLan, 155, 156
rewrite, 45

safe query language, 136
same as, *see* `owl:sameAs`
satisfiability
ABox, 221
concept, 176, 210, 212, 215, 222, 249, 254, 268, 272, 281, 287, 302, 312, 313, 346, 379, 385
for set of ABoxes, 262
`satisfiable`, $\mathcal{ALCN}$ tableau function, 313
satisfiable node, 400

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

schema, *see* RDF, schema; XML, schema
search
  engine, 21–48, 122
  space, 144, 276, 284, 399
  tree, 271, 275, 276, 347, 399
security problems, 19–20
see also, *see* `rdfs:seeAlso`
`select`, $\mathcal{ALCN}$ tableau function, 342
self, *see* `owl:hasSelf`
self-realising ABox, 279, 280, 289, 351, 352, 356, 361, 374, 385
self restriction, 240
semantic branching, 332, 396
semantic consequence, 188, 220
Semantic Web, 52, 66, 98
semantics, 41, 45, 47, 53, 65, 185
  ABox, 218
  $\mathcal{AL}$, 184, 186
  $\mathcal{ALC}$, 189
  $\mathcal{ALCN}$, 195
  RDF, 70
  $\mathcal{SHIQ}$, 203
  $\mathcal{SROIQ}$, 242
sequence, *see* `rdf:Seq`
serialisation, 141
server, 9, 12, 14, 19
servlet, 21, 131
session, 20
SGML, 64
  specification, 55
shared subject, 75
$\mathcal{SHIQ}$ language, 197, 203, 227, 375
$\mathcal{SHIQ}$ tableau state, 378
$\mathcal{SHOIN}$ language, 447
signature, 228, 230
similarity measure, 26
simple role, 441
  in $\mathcal{SHIQ}$, 202
  in $\mathcal{SROIQ}$, 241
simulated universal role, 214
sink, 34
SINTAGMA, 155
SOAP protocol, 58, 154
some values from, *see* `owl:someValuesFrom`
`sort`, Haskell function, 339
soundness of tableau algorithm, 278–281, 288–290, 297, 383–385
`Source`, Dublin Core, 111
source individual, *see* `owl:sourceIndividual`
SPARQL, 151–155
  specification, 153
specification
  CGI, 14
  HTML, 10, 53
  HTTP, 15
  OWL 1, 408
  OWL 2, 408
  RDF, 66
  RDF schema, 98
  RQS, 136
  SGML, 55
  SPARQL, 153
  XML, 55
  XML schema, 62, 96
  XPath, 133
spider, web, 22, 39, 40, 43, 125
spiral, 372
SQL, 130, 135, 139, 147, 151, 156
$\mathcal{SROIQ}$ language, 239, 447
stable node, 286, 287, 298
state, *see* tableau, state
statement, *see* `rdf:Statement`
static blocking, 285, 286
static page, 11–12, 14
stochastic surfer model, 31, 34
structural subsumption algorithm, 211, 250–254
  minimality principle, 250, 253
stylesheet, 64
  cascading, CSS, 62
  XSLT, 62
sub (subconcepts function), 267
subclass, 99, 101, 427
subclass of, *see* `rdfs:subClassOf`
subcomponent, 197
subconcept, 191, 267
subgraph matching, 141, 143, 144, 151
`Subject`, Dublin Core, 110
subject, *see* `rdf:subject`; RDF, subject
sublanguage, 189, 193, *see also* OWL, sublanguages
subneg (negated subconcepts function), 377
subproperty of, *see* `rdfs:subPropertyOf`
subset blocking, 285, 337, 351, 400
substitution, 141, 143, 144
`subsumes`, $\mathcal{ALCN}$ tableau function, 314
subsumption, 181, 428, *see also* structural subsumption algorithm
  of concepts (TBox reasoning), 250
  of concepts (TBox reasoning), 176, 210, 250, 314, 401
subtype, 232
`successors`, $\mathcal{ALCN}$ tableau function, 327
successor node, 250, 256, 286, 358, 360, 369, 378, 388
superclass, 413
surface web, 38
symbol, 207, 209, 212, 228, 395
symmetric property, *see* `owl:SymmetricProperty`
symmetric role, 200, 206, 240, 440
symmetry, 99, 440
syndication, 115, 117
synonym, 26, 91, 117
syntax

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

ABox, 217
$\mathcal{AL}$, 183
$\mathcal{ALC}$, 189
$\mathcal{ALCN}$, 195
RDF Notation 3, 84
RDF XML, 73–84
$\mathcal{SHIQ}$, 203
$\mathcal{SROIQ}$, 240

`Tableau`, $\mathcal{ALCN}$ tableau datatype, 307
tableau
algorithm, 203, 211, 214, 304–387
ABox, 291–299, 385–387
$\mathcal{ALCN}$, 267–281
for non-empty TBox, 286–291
iterative, 272, 381
minimality principle, 257, 261, 270, 290
optimisation, 387–404
recursive, 271
$\mathcal{SHIQ}$, 375–385
graph, 224, 255, 268, 306, 346, 378
rule, 256, 269, 278, 314, 379, 382, 388
$\bowtie$-, 375, 380
$\exists$-, 257, 269, 287, 332, 361, 370, 379
$\forall$-, 269, 322, 355, 361, 379
$\forall_+$-, 257, 349, 350, 355, 361, 379
$\geqslant$-, 266, 269, 287, 335, 370, 380
$\leqslant$-, 263, 266, 269, 326, 370, 374, 380, 386
$\sqcap$-, 256, 269, 320, 361, 379
$\sqcup$-, 258, 269, 331, 361, 379
expansion, 270, 286, 292, 317, 359, 360
firing condition, 270, 314, 379
non-deterministic, 258, 261, 263, 266, 270, 292, 315, 317, 326, 331, 375, 381, 387
parameter, 270, 381
state, 256, 266, 268, 306, 378
complete, 257, 270–272, 280, 289, 346, 381
unification, 298
transformation, *see* tableau, rule
`tableau`, $\mathcal{ALCN}$ tableau function, 315
`TableauTransformation`, $\mathcal{ALCN}$ tableau type, 314, 321
tail of Haskell list, 315
`tails`, Haskell function, 319
target individual, *see* `owl:targetIndividual`
Tarski semantics, 185
TBox (terminological box), 176, 206, 281, 286, 311, 392, 407
acyclic, 207, 209, 212–213, 223, 393–395
cyclic, 207, 281
definitional, 207
definitional part of, 207
definitorial, 207–209
expansion of, 208, 212
TBox reasoning
classification, 177, 200, 211, 401
coherence, 210, 403
concept disjointness, 176, 210
concept equivalence, 210
concept satisfiability, 176, 210, 212, 215, 222, 249, 254, 268, 272, 281, 287, 302, 312, 313, 346, 379, 385
concept subsumption, 176, 210, 250, 314, 401
consistency check, 210, 212
`tboxConcept`, $\mathcal{ALCN}$ tableau selector function, 307
`tboxInternalise`, $\mathcal{ALCN}$ tableau function, 311
termination of tableau algorithm, 275–277, 284, 288, 381–382
terminological axiom, 176, 179, 212–217, 281, 392
concept axiom, 181, 187, 206
role axiom, 181, 206
terminological box, *see* TBox
terminological system, 66, 176, 206
thing, *see* `owl:Thing`
`Title`, Dublin Core, 110
top, 179, 183, 186, 203, 204, 305, 401, 413, *see also* `owl:Top`
top object property, *see* `owl:topObjectProperty`
transformation
of tableaux, *see* tableau, rule
XML, 62
transitive closure, 207, 238, 286
weak, 199
transitive property, *see* `owl:TransitiveProperty`
transitive role, 197, 198, 202, 203, 205, 223, 240, 348–352
transitivity, 99, 440
transitivity closure, 352
transitivity rule for value restriction, 257, 349, 350, 355, 361, 379
triple (RDF), 70
type, *see* `rdf:type`
`Type`, Dublin Core, 111
type synonym, Haskell, 306
type variable, Haskell, 339

UNA, *see* unique name assumption
unblocking, *see* blocking
undecidable logic, *see* decidable logic
unfolding transformation, 256, 258, 320, 331, 390, 392, 395
unified tableau, 298
Uniform Resource Identifier (URI), 59, 66
absolute, 67, 75
base, 67, 78, 81
fragment identifier #, 67, 69, 71, 75, 78, 96
relative, 67, 78, 81
Uniform Resource Locator (URL), 11, 66
Uniform Resource Name (URN), 66

© in this web service Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-70036-8 - The Semantic Web Explained: The Technology and Mathematics behind Web 3.0
Péter Szeredi, Gergely Lukácsy and Tamás Benkő
Index
More information

union, 180, 189, 203, 204, 234, 258, 389, 391, 392, 397–399, 424–425
union rule, 258, 269, 331, 361, 379, 396
union of, *see* `owl:unionOf`
`unique`, $\mathcal{ALCN}$ tableau function, 339
unique name assumption, 219, 230, 234, 261, 263, 292, 386, 411, 421, 441, 442
universal
  concept, 179, 183, 186, 203, 204, 305, 413
  name, 59
  property, 448
  restriction, 180, 416
  role, 214, 234, 240, 448
Universally Unique Identifier (UUID), 67
universe, 185
unnamed class, 413, 431, 432
unqualified number restriction, 195, 265
unsatisfiability, *see* satisfiability
URI, *see* Uniform Resource Identifier
URL, *see* Uniform Resource Locator
URN, *see* Uniform Resource Name
`usort`, $\mathcal{ALCN}$ tableau function, 321, 327, 340
UUID, *see* Universally Unique Identifier

value, *see* `owl:allValuesFrom`; `owl:hasValue`; `rdf:value`
value constraint
  OWL, 415–419
  SPARQL, 152
value restriction, 180, 183, 186, 203, 204, 417
  rule for, 269, 322, 355, 361, 379
  transitivity rule for, 257, 349, 350, 355, 361, 379
variable
  environment variable, 16, 18
  first-order logic, 227
  Haskell type, 339
  RDFLan, 155
  RDF graph pattern, 141, 142
  RDQL, 147
  substitution, 141, 143, 144
  XQuery, 131, 134
vector space model (VSM), 25, 26

weak transitive closure, 199
web, 9–42, 44, 52, 54, 66, 98
Web Ontology Language, *see* OWL
web page, 9
  dynamic, 12–14, 17, 38
  static, 11–12, 14
web robot, 22, 39, 40, 43, 125
web site, 33–37
web spider, 22, 39, 40, 43, 125
`where`, Haskell keyword, 312, 341
Wordnet, 117
World Wide Web, 9–42
World Wide Web Consortium (W3C), 55, 131, 151
wrapper, 44

XHTML, 64, 123
`xml:base` attribute, 78, 82
XML, 55–65, 68, 73, 130
  attribute, 56, 58
  element, 56
  entity, 60
  hierarchy, 56
  namespace, 59, 74, 100
  qualified name, 59, 74, 75, 78
  query languages, 130, 135
  schema, 61, 62, 96
  specification, 55
  tag, 56
  transformation, 62
  wellformedness, 60
XPath, 131–133
  specification, 133
XQuery, 131, 133
XSL, 62
XSLT, 62

© in this web service Cambridge University Press www.cambridge.org