

Index

- acceptance
 - of alternating tree automaton, 118
 - of string automaton, 12, 178
 - of tree automaton, 33
 - of tree-walking automaton, 152
- accumulating parameter, *see* parameter
- all-matches semantics, 50, 90
- alternating run, 118
- alternating tree automaton, 4, 117–126, 141, 153
- alternating tree language, 118
- Ambient Logic, 175
- ambiguity, 4, 50–54, 176–189
 - of string automaton, 178
 - strong, 177
 - weak, 178
- assignment
 - first-order, 162
 - second-order, 165
- assumption set, 113
- ATL**, class of alternating tree languages, 118
- attribute, 21, 190–194
- attribute-element automaton, 196
- attribute-element constraint, 193
- axis, 148
- backtracking, 94, 114
- backward inference, 6, 72, 141
- binarization, 36
- binary decision diagram, 117, 175, 198
- binary tree, *see* tree
- binding, 50, 57, 74, 99, 186
 - linear, 57
- binding ambiguity, 186
- boolean attribute grammar, 117
- BU**, class of languages accepted by bottom-up deterministic tree automata, 42
- C#, 90
- call-by-name semantics, 136
- call-by-value semantics, 136
- canonical form
 - of MSO formula, 168
 - of pattern, 68
 - of schema, 37
 - of type, 37
- caterpillar atom, 150
- caterpillar expression, 150, 160, 164
- caterpillar path, 150
- chain of references, 25
- choice
 - of attributes, 191
 - pattern, 57
 - type, 23
- complementation
 - of alternating tree automaton, 124
 - of string automaton, 16
 - of tree automaton, 44
- composability, 137
- concatenation, 9
 - pattern, 57
 - type, 22
- conclusion predicate, 12
- conformance relation, 11, 25, 74, 177
- conjunction, 162
- conjunctive path expression, 149
- construction
 - of alternating tree automaton from tree-walking automaton, 157
 - of marking tree automaton from MSO formula, 170
 - of marking tree automaton from pattern, 66
 - of sequence-marking tree automaton from pattern, 69
 - of tree automaton from schema, 37
- containment algorithm, 109–116
 - bottom-up, 109
 - top-down, 110–116
- content, 21
- content model, 24
- content type, 24
- Context Logic, 175
- context-free grammar, 25

context-free tree language, 72, 140, 141
 currification, 103

data model, 19–22
 denotation function, 128, 132
 derivation, 12
 descendant, 32
 determinization, 42, 95, 121
 difference automaton, 109, 111
 disjunction, 162
 disjunctive normal form, 119
 divide and conquer, 196
DNF, *see* disjunctive normal form
 DocBook, 2
 document, 1, 19
 document order, 32, 163
 document type definition (DTD), 1, 27, 41, 184, 197

ϵ -elimination, 13, 36, 66
 ϵ -normal form condition, 181
 ϵ -transition, 13, 35, 66
 elaborated label, 176
 elaboration, 176
 element, 19
 element consistency, 27
 emptiness test algorithm, 45
 for alternating tree automaton, 125
 for string automaton, 16
 top-down, 112
 empty sequence, 9
 pattern, 57
 type, 22
 value, 21
 epsilon normal form, 181
eval function, 106
 evaluation relation, 74
 exhaustiveness, 78
 existential quantification
 first-order, 162
 second-order, 165
 expression, 74

family tree, 19
 family tree schema, 24
 far dependency, 28
first, set of labels, 179
 first-order logic, 5, 161–165
 flow analysis, 90
follow, set of labels, 179
 forest transducer, 136
 macro, 136, 147
 formula
 for alternating tree automaton, *see* transition
 formula
 for first-order logic, 162
 for monadic second-order logic, 165
 forward inference, 72, 140
 function call, 74
 function definition, 74
 function name, 74

Glushkov automaton, 179
 grammar, 10
 greedy match, 53, 59, 68

Haskell, 90
 height, 31
 height property, 130
 of macro tree transducer, 134
 of top-town tree transducer, 131
 higher-order function, 90
 HTML, 1

image, 140
 implication, 162
 inference rule, 12
 interleave, 194
 International Standards Organization (ISO), 2
 intersection
 of string automata, 16
 of tree automata, 44
 intersection type, 4, 119
 irredundancy, 78
 iterator, 56

Java, 2, 90
 J Wig, 90

label, label of node, 31
 label, 9, 19
 pattern, 57
 type, 22
 language
 of string automaton, 12
 of regular expression, 11
last, set of labels, 179
leaves, set of leaves, 31
 lexicographic order, 9
 logic, 5

marking, 63
 linear, 65
 marking algorithm, 99–108
 bottom-up, 102
 top-down, 100
 marking run
 bottom-up, 64
 top-down, 64
 marking string automaton, 186
 marking tree automaton, 5, 63
 linear, 65, 99, 174
 marking ambiguity, 187
 matching of marking tree automaton, 64
 matching relation, 58, 74
 prioritized, 59
 MathML, 1
 McNaughton–Yamada construction, 14, 68
 membership algorithm, 93–99
 bottom-up, 95
 with top-down filtering, 97

- for alternating tree automaton, 123
 - top-down, 94
- model, 161
- MONA, 175, 178
- monadic second-order logic (MSO), 5, 117, 165
- MTran*, 175
- μ XDuce, 6, 73–82, 127, 138
- name space, 22
- ND**, class of regular tree languages, 33, 40, 43, 120, 153
- negation, 162
- node, 19, 31
- node kind, 152, 160
- node test, 148
- nodes**, set of nodes of a tree, 31
- nondeterministic match, 53, 74
- non-tail sequence, 67
- O’Caml, 90
- object orientation, 2
- OcamlDuce, 90
- on-the-fly algorithm, 4, 93–117, 174
- one-unambiguity, 41, 184
- Organization for the Advancement of Structured Information Standards (OASIS), 11
- open attribute, 191
- optional attribute, 191
- optional type, 23
- ordering, 21, 190
- parameter, 132
- parametric polymorphism, 90, 189
- partial derivative, 70, 198
- partially lazy multiset operation, 102, 105
- path, 148
- path expression, 5, 148–152, 163
- pattern, 5, 49, 56, 74, 163, 186
 - linear, 54–56, 59, 74
- pattern definition, 52, 57, 74
- pattern match, 74
- pattern matching, 49–62
- pattern name, 56
- pattern schema, 57
- pos**, set of labels, 179
- position, 31
- precision of typechecking, 71
- prefix, 9
- preimage, 141
- premise, 12
- Presburger’s arithmetics, 198
- primitive binary relation, 162
- primitive unary relation, 162
- prioritized match, 53, 90
- priority id, 59
- procedure, 127
- product construction, 16, 44, 197
- product tree automaton, 44, 83
- program, 74
- progress theorem, 81
- proper analysis, 167
- query on a tree, logic formula as, 162
- regular expression, 2, 10, 22
- regular expression path, 150, 160
- regular look ahead, 136
- regular string language, 3, 12
- regular tree language, 3, 33, 140
- RELAX NG, 2, 27, 195
- repetition
 - pattern, 57
 - type, 23
- run, 12, 32, 152, 178
 - bottom-up, 34, 97
 - top-down, 33, 97
- satisfaction judgment, 163, 166
- scalable vector graphics (SVG), 11
- schema, 1, 19–29
 - local, 27, 41
 - regular, 27
 - single, 27, 41
- schema language, 1, 19
- schema model, 4, 19–29
- self-or-descendant relation, 32, 163
- semantics
 - of μ XDuce, 74
 - of alternating tree automaton, 117
 - of attribute-element constraint, 193
 - of caterpillar expression, 150
 - of first-order logic, 163
 - of macro tree transducer, 132
 - of marking tree automaton, 63
 - of monadic second-order logic, 166
 - of pattern, 57
 - of regular expression, 10
 - of schema, 25
 - of string automaton, 12
 - of top-down tree transducer, 128
 - of tree automaton with ϵ -transition, 35
 - of tree automaton, 32
 - of tree-walking automaton, 152
- sequence, 9
- sequence-capturing semantics, 56, 69, 89
- sequence-linear marking, 69
- sequence-marking tree automaton, 69, 89
- Sheaves Logic, 175
- shuffle automaton, 197
- shuffle expression, 194–196
- simple object access protocol (SOAP), 11
- single-match semantics, 50
- standard generalized markup language (SGML), 12
- star normal form (SNF), 181
- star normal form condition, 183
- state, 12, 32
 - destination, 32
 - final, 12, 32
 - initial, 12, 32
 - reachable, 47
 - source, 32

- state (*cont.*)
 - useful, 47
 - useless, 47
- state sharing, 116
- stay rule, 136
- string, 19
- string automaton, 12
 - deterministic, 15
- String, type, 22
- subset construction, 16, 42, 95
- subset tree automaton, 42
- subtree**, subtree at a particular node, 31
- subtyping relation, 76
- suffix, 9
- tag, 19
- tail sequence, 67
- TD**, class of languages accepted by
 - top-down tree automata, 40
- tour, 153
 - reduced, 156
 - simple, 155
 - trivial, 154
- transition formula, 117
- transition rule, 12, 32, 63
- tree, 30
- tree automaton, 3, 25, 30–48
 - bottom-up deterministic, 42, 121
 - complete, 42
 - nondeterministic, 32, 120
 - top-down deterministic, 40, 95
 - with ϵ -transition, 35
- tree transducer, 6, 127–137
 - k -pebble, 137, 147
 - bottom-up, 136, 147
 - high-level, 136, 147
 - macro, 131, 147
 - MSO-definable, 137, 174
 - top-down, 6, 127, 138, 147
 - deterministic, 128
 - linear, 141
 - total, 128
- tree-walking automaton, 5, 152–160
 - alternating, 159
 - deterministic, 159
- Turing machine, 6, 71
- TWA**, class of languages accepted by
 - tree-walking automata, 152
- type, 3, 22, 74
- type annotation, 73
- type definition, 24, 74
- type environment, 76
- type error, 71, 80
- type inference for pattern, 78, 82, 139
- type name, 23
- type preservation theorem, 80
- type system, 73
- typechecking, 3, 71–90
 - approximate, 5, 72, 140
 - exact, 5, 72, 138–147
- typing relation, 76
- typing rule, 73
- unambiguous match, 53, 90
- unelaboration, 176
- ungreedy match, 53
- union
 - of string automata, 16
 - of tree automata, 44
- universal quantification
 - first-order, 162
 - second-order, 165
- unorderedness, 4, 190–198
- unranked tree, 29
- useless-state elimination, 47, 83, 184, 187
- value, 20, 74
- variable, 49, 63
 - first-order, 162
 - non-tail, 67, 88
 - second-order, 165
 - tail, 67
- variable binder, 49
- variable unary relation, 165
- weak second-order logic with two successors
 - (WS2S), 165
- well-formedness, 25, 37, 57
- wildcard, 164
- World Wide Web Consortium (W3C), 11
- XAct, 90
- XDuce, 6, 71, 90
- XHaskell, 90
- XHTML, 1, 133, 164
- XJ, 90
- XML, 1
- XML Schema, 1, 27, 41, 184, 196
- XML typechecking problem, 72
- XPath, 5, 149, 160
- XSLT, 90, 127
- Xtatic, 90