DOCUMENTATION OF SPARQL QUERIES IN MLT-OWL

Technical Report, v1.0, June 2016.

Freddy Brasileiro[a], João Paulo A. Almeida[a], Victorio A. Carvalho[a,b] and Giancarlo Guizzardi[a]

[a]Ontology & Conceptual Modeling Research Group (NEMO), Federal University of Espírito Santo (UFES), Vitória, ES, Brazil.

[b]Research Group in Applied Informatics, Informatics Department, Federal Institute of Espírito Santo (IFES), Colatina, ES, Brazil.

freddybrasileiro@gmail.com, jpalmeida@ieee.org, victorio@ifes.edu.br, gguizzardi@inf.ufes.br

CONTENTS

LIST OF TABLES

1. DERIVATION RULES

The following tables describe the derivation rules for MLT-OWL. The queries are described in SPARQL, with a corresponding action, which is executed whenever the query returns non-empty results. The MLT axioms and theorems that motivate each derivation rules are indicated.

Table 1. Derivation Rules Corresponding to Axiom A3 of MLT

SPARQL query	Action

`SELECT DISTINCT ?x`
`WHERE {`
`  ?t rdf:type mlt:1stOrderClass .`
`  ?x rdf:type ?t .`
`}`
	For each returned ?x, add that
	?x is an instance of *mlt:TokenIndividual*

`SELECT DISTINCT ?t`
`WHERE {`
`  ?x rdf:type ?t .`
`  ?x rdf:type mlt:TokenIndividual .`
`  filter(?t != mlt:TokenIndividual) .`
`}`
	For each returned ?t, add that
	?t is an instance of *mlt:1stOrderClass*

Table 2. Derivation Rules Corresponding to Axiom A4 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t1`
`WHERE {`
`  ?t rdf:type mlt:2ndOrderClass .`
`  ?t1 rdf:type ?t .`
`}`
	For each returned ?t1, add that
	?t1 is an instance of *mlt:1stOrderClass*

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t1 rdf:type ?t .`
`  ?t1 rdf:type mlt:1stOrderClass .`
`  filter(?t != mlt:TokenIndividual) .`
`}`
	For each returned ?t, add that
	?t is an instance of *mlt:2ndOrderClass*

Table 3. Derivation Rules Corresponding to Axiom A5 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t1`
`WHERE {`
`  ?t rdf:type mlt:3rdOrderClass .`
`  ?t1 rdf:type ?t .`
`}`
	For each returned ?t1, add that
	?t1 is an instance of *mlt:2ndOrderClass*

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t1 rdf:type ?t .`
`  ?t1 rdf:type mlt:2ndOrderClass .`
`  filter(?t != mlt:TokenIndividual) .`
`}`
	For each returned ?t, add that
	?t is an instance of *mlt:3rdOrderClass*

Table 4. Derivation Rule Corresponding to Axiom D1 of MLT

SPARQL query	Action

`SELECT DISTINCT ?e ?t2`
`WHERE {`
`  ?t1 rdfs:subClassOf+ ?t2 .`
`  ?e rdf:type ?t1`
`}`
	For each pair ?e and ?t2, add that ?e is an instance of ?t2

Table 5. Derivation Rules Corresponding to Axiom D4 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t2 ?t3`
`WHERE {`
`  ?t1 mlt:isPowertypeOf ?t2 .`
`  ?t3 rdf:type ?t1 .`
`  ?t1 rdf:type ?t1Type .`
`  filter(?t1Type != mlt:TokenIndividual) .`
`}`
	For each pair ?t2 and ?t3, add that ?t3 is a subclass of ?t2

`SELECT DISTINCT ?t1 ?t3`
`WHERE {`
`  ?t1 mlt:isPowertypeOf ?t2 .`
`  ?t3 rdfs:subClassOf* ?t2 .`
`  ?t1 rdf:type ?t1Type .`
`  filter(?t1Type != mlt:TokenIndividual) .`
`}`
	For each pair?t1 and ?t3, add that ?t3 is an instance of ?t1

Table 6. Derivation Rule Corresponding to Axiom D5 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t2 ?t3`
`WHERE {`
`  ?t1 mlt:characterizes ?t2 .`
`  ?t3 rdf:type ?t1 .`
`  ?t1 rdf:type ?t1Type .`
`  filter(?t1Type != mlt:TokenIndividual) .`
`}`
	For each pair?t2 and ?t3, add that ?t3 is a subclass of ?t2

Table 7. Derivation Rule Corresponding to Axiom D6 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t1 ?t2`
`WHERE {`
`  ?t1 mlt:completelyCharacterizes ?t2 .`
`}`
	For each pair of ?t1 and ?t2, add that [?t1, mlt:caracterizes, ?t2]

Table 8. Derivation Rule Corresponding to Axiom D7 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t1 ?t2`
`WHERE {`
`  ?t1 mlt:disjointlyCharacterizes ?t2 .`
`}`
	For each pair ?t1 and ?t2, add that [?t1, mlt:caracterizes, ?t2]

Table 9. Derivation Rules Corresponding to Axiom D8 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t1 ?t2`
`WHERE {`
`  ?t1 mlt:partitions ?t2 .`
`}`
	For each pair ?t1 and ?t2, add that [?t1, completelyCharacterizes, ?t2] [?t1, disjointlyCharacterizes, ?t2]

`SELECT DISTINCT ?t1 ?t2`
`WHERE {`
`  ?t1 mlt:completelyCharacterizes ?t2 .`
`  ?t1 mlt:disjointlyCharacterizes ?t2 .`
`}`
	For each pair ?t1 and ?t2, add that [?t1, mlt:partitions, ?t2]

Table 10. Derivation Rules Corresponding to Theorem T7 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t rdf:type mlt:1stOrderClass .`
`}`
	For each returned ?t, add that ?t is a subclass of *mlt:TokenIndividual*

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t rdfs:subClassOf+ mlt:TokenIndividual.`
`}`
	For each returned ?t, add that ?t is a subclass of *mlt:1stOrderClass*

Table 11. Derivation Rules Corresponding to Theorem T8 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t rdf:type mlt:2ndOrderClass .`
`}`
	For each returned ?t, add that ?t is a subclass of *mlt:1stOrderClasss*

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t rdfs:subClassOf+ mlt:1stOrderClass .`
`}`
	For each returned ?t, add that ?t is a subclass of *mlt:2ndOrderClass*

Table 12. Derivation Rules Corresponding to Theorem T9 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t rdf:type mlt:3rdOrderClass .`
`}`
	For each returned ?t, add that
	?t is a subclass of *mlt:2ndOrderClasss*

`SELECT DISTINCT ?t`
`WHERE {`
`  ?t rdfs:subClassOf+ mlt:2ndOrderClass .`
`}`
	For each returned ?t, add that
	?t is a subclass of *mlt:3rdOrderClass*

Table 13. Derivation Rule Corresponding to Theorem T15 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t3 ?t4`
`WHERE {`
`  ?t2 rdfs:subClassOf+ ?t1 .`
`  ?t4 mlt:isPowertypeOf ?t2 .`
`  ?t3 mlt:isPowertypeOf ?t1 .`
`}`
	For each pair of ?t3 and ?t4, add that
	?t4 is a subclass of ?t3

Table 14. Derivation Rule Corresponding to Theorem T16 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t1 ?t3`
`WHERE {`
`  ?t1 mlt:isSubordinateTo ?t2 .`
`  ?t2 mlt:characterizes ?t3 .`
`}`
	For each pair of ?t1 and ?t3, add that
	[?t1, mlt:characterizes, ?t3]

Table 15. Derivation Rule Corresponding to Theorem T17 of MLT

SPARQL query	Action

`SELECT DISTINCT ?t2 ?t3`
`WHERE {`
`  ?t2 mlt:isPowertypeOf ?t1 .`
`  ?t3 mlt:characterizes ?t1 .`
`}`
	For each pair of ?t2 and ?t3, add that
	?t3 is a subclass of ?t2

2. INTEGRITY CONSTRAINTS

The following tables describe integrity constraints for MLT-OWL, implemented as SPARQL queries. The MLT axioms and theorems that motivate each constraint are indicated. Violations ot the MLT rules are detected whenever the query returns non-empty results.

Table 16. Integrity Constraint Corresponding to Axiom A1 of MLT

SPARQL query

`SELECT DISTINCT *`
`WHERE {`
`  ?x rdf:type mlt:TokenIndividual .`
`  ?y rdf:type ?x .`
`}`

Table 17. Integrity Constraint Corresponding to Theorem T5 of MLT

SPARQL query

`SELECT DISTINCT *`
`WHERE {`
`  ?x rdf:type ?y .`
`  ?y rdf:type ?x .`
`}`

Table 18. Integrity Constraint Corresponding to Theorem T6 of MLT

SPARQL query

`SELECT DISTINCT *`
`WHERE {`
`  ?x rdf:type ?y .`
`  ?y rdf:type ?z .`
`  ?x rdf:type ?z .`
`}`

Table 19. Integrity Constraint Corresponding to Axiom T13 of MLT

SPARQL query

`SELECT DISTINCT *`
`WHERE {`
`  ?p mlt:isPowertypeOf ?t .`
`  ?p1 mlt:isPowertypeOf ?t .`
`  FILTER (?p NOT IN (?p1)) .`
`}`

Table 20. Integrity Constraint Corresponding to Axiom T14 of MLT

SPARQL query

SELECT DISTINCT *
WHERE {
?p mlt:isPowertypeOf ?t .
?p mlt:isPowertypeOf ?t1 .
FILTER (?t NOT IN (?t1)) .
}

Table 21. Integrity Constraint Corresponding to Axiom T18 of MLT

SPARQL query

SELECT DISTINCT *
WHERE {
?t1 mlt:partitions ?t3 .
?t2 mlt:partitions ?t3 .
?t1 mlt:properSpecializes ?t2 .
}