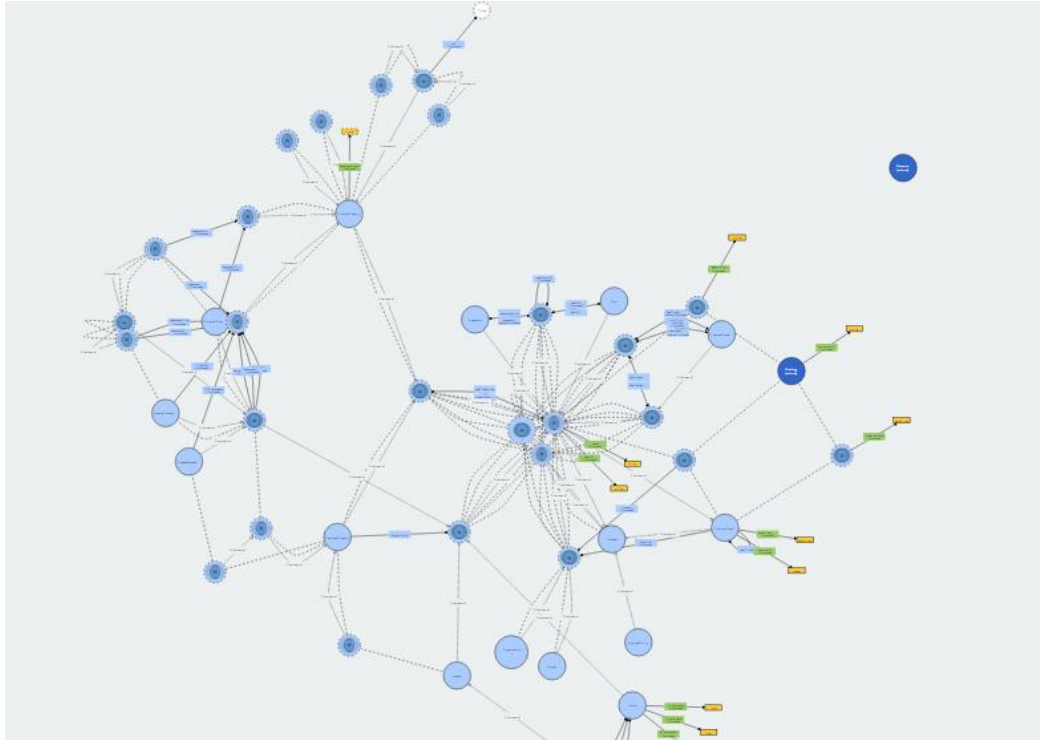


COINS 2.0 Lite & Lite+eq Profielen



04-07-2016

Versie: 0.96

Dr. ir. Hans Schevers

Inhouds opgave

1	Inleiding	3
2	Introductie OWL semantiek	4
2.1	Inleiding	4
2.2	OWL semantiek	4
2.3	Open World en Closed World.....	4
2.3.1	Open World Aannamen.....	4
2.3.2	Closed World Aannamen.....	5
2.3.3	Unique Naming Assumption.....	5
3	Wat is COINS Semantiek?	6
3.1	Semantisch BIM model.....	6
3.2	Schema nivo afleidingen.....	6
3.3	Instance semantiek.....	7
3.4	Afleidingen en validatie	7
3.5	Ongewenste OWA OWL afleidingen	8
3.6	Conclusies.....	8
4	Coins 2.0 Lite en COINS 2.0 Lite+eq profielen	10
4.1	2 profielen	10
4.2	COINS semantiek overzicht	10
4.2.1	Cardinaliteits restricties.....	10
4.2.2	Functionalproperty.....	10
4.2.3	Domain en Range van owl:properties	10
4.2.4	Union class.....	10
4.2.5	ExpiredEntity	10
4.3	COINS Lite vocabulair	10
4.3.1	COINS 2.0 Lite vocabulair	10
4.3.2	COINS 2.0 Lite+Eq vocabulair	12
4.4	COINS Referentie kader en COINS OTL profiel testen.....	12
4.4.1	COINS 2.0 Lite profiel test	12
4.4.2	COINS lite+eq profiel test	13
5	COINS 2.0 Lite en Lite+Eq Semantiek	14
5.1	Inleiding	14
5.2	Schema afleidingen	14
5.3	Data afleidingen	15
5.4	Validatie.....	16
5.5	Executie van de Sparql queries.....	17
6	Naar een COINS 2.0 validatie service & tool	18
6.1	Prototype implementatie	18
6.2	COINS 2.0 API uitbreiding.....	18
6.3	Performance	18
6.4	Aanvullende validaties voor een COINS 2.0 Validatie service	18

6.5	Extra eisen voor een Validatie Service	18
7	Bijlage 1: Schema Entailment Sparql queries	20
8	Bijlage 2: Instance Entailment Sparql queries	24
9	Bijlage 3: Data Validation Sparql queries	27

1 Inleiding

In dit document wordt de COINS 2.0 semantiek geïntroduceerd waarmee geformaliseerd wordt welke afleidingen en validaties op datasets mogelijk zijn. De semantiek is vast gelegd mbv 2 profielen te weten COINS 2.0 Lite en COINS 2.0 Lite+Eq. De 2 profielen worden geformaliseerd door een set van Sparql queries die de daadwerkelijke formalisatie van de semantiek beschrijven. In de bijlagen zijn alle Sparql queries te vinden.

2 Introductie OWL semantiek

2.1 Inleiding

In dit hoofdstuk wordt er meer inzicht geboden in de werking van de Web Ontology Language (OWL) en wordt globaal het verschil uitgelegd tussen Open World Aannamen (OWA) en Closed World Aannamen (CWA). Dit is relevant aangezien COINS werkt op basis van CWA terwijl de onderliggende techniek van COINS, te weten RDF/OWL, gebaseerd is op OWA.

2.2 OWL semantiek

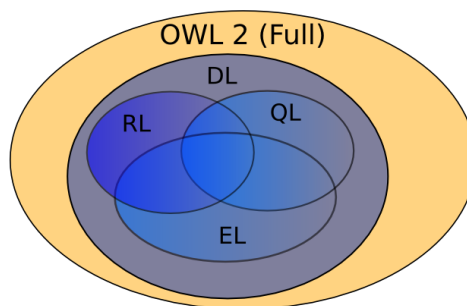
Er zijn twee OWL 2.0 semantiekken gedefinieerd door de W3C te weten:

- 1) OWL Direct Semantics (<https://www.w3.org/TR/owl2-direct-semantics/>)
- 2) OWL RDF based Semantics (<https://www.w3.org/TR/owl2-rdf-based-semantics/>)

Beide definities gaan uit van een zogenaamde Open World Aannamen (OWA). De Direct Semantics kan gezien worden als een subset van de RDF-based Semantics.

Beide definities zijn uitgebreid en kunnen daardoor veel computer resources vergen waardoor er zogenaamde profielen zijn ontstaan om een optimum te vinden tussen de gevraagde resources en de ondersteuning van de semantiek (zie <https://www.w3.org/TR/owl2-profiles/> voor meer informatie over de aanezige OWL profielen).

Deze profielen zijn subsets van de totale OWL semantiek en kunnen daardoor minder computer resources vragen en ondersteunen daardoor ook niet alles. De hoeveelheid complexiteit en schaalbaarheid hebben o.a. invloed gehad op deze de profielen. Grofweg is het volledige OWL Full (RDF semantics) minder schaalbaar dan bijvoorbeeld het QL (QueryLanguage) profiel van OWL. Onderstaande figuur laat de verschillende profielen zien die momenteel aanwezig zijn zoals OWL-Rule Language (RL), OWL-EL, OWL-Query Language (QL) en OWL-Description Logic (DL).



2.3 Open World en Closed World

2.3.1 Open World Aannamen

OWL is gebaseerd op Open World Aannamen. Grofweg houdt dit in dat wanneer er geen informatie is dat er geen gevolgtrekking kan zijn. Oftewel gevolgtrekking is altijd op basis van bestaande informatie waarbij de de gevolgtrekking ook monotoon is. Dat wil zeggen dat het invoegen van nieuwe informatie (nieuwe triples) oude gevolgtrekking altijd 'waar' blijft. Het kan dus niet zo zijn dat door het invoegen van nieuwe triples er oude gevolgtrekkingen ongedaan worden of onjuist zijn. Wel kan er natuurlijk extra gevolgtrekkingen mogelijk zijn door het invoegen van nieuwe triples.

2.3.2 Closed World Aannamen

De Closed World Aannamen (CWA) biedt grofweg wel de mogelijkheid om gevolgtrekkingen te laten plaats vinden op basis van ‘missende’ informatie. Wat onbekend is (missende informatie) kan beschouwd worden als ‘niet waar’. Hiermee is de gevolgtrekking niet monotoon. Afleidingen op basis van missende informatie zijn immers niet meer valide wanneer deze informatie wel wordt toegevoegd.

2.3.3 Unique Naming Assumption

Een belangrijk onderliggend principe van CWA is de Unique Naming Assumption (UNA). Elke resource is standaard uniek en dus anders dan andere resources. De Open World Assumption gaat niet uit van UNA. In OWA zou dit wel gemodelleerd kunnen worden door de ‘differentFrom’ relatie te leggen tussen resources. Daarmee is expliciet UNA te modelleren. CWA gaat er dus wel vanuit dat elke resource uniek is en hoeven deze ‘differentFrom’ relaties niet gelegd te worden.

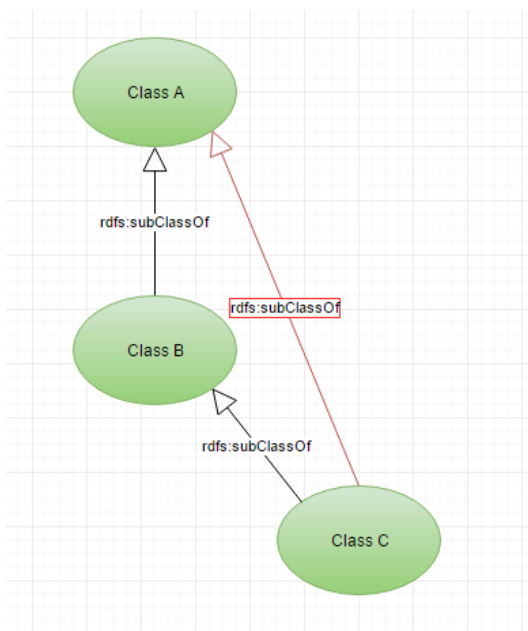
3 Wat is COINS Semantiek?

3.1 Semantisch BIM model

Het Semantisch BIM model van COINS 2.0 zijn RDF/OWL datasets met referenties naar COINS 2.0 Core model, Referentie kaders en OTL's. Je zou het Semantisch BIM als een 'data' ontologie kunnen beschouwen en ontologieën waarnaar gerefereerd worden de 'Schema' ontologieën.

3.2 Schema nivo afleidingen

Op schema nivo zijn er veel gevolgtrekkingen mogelijk. Een voorbeeld is de specialisatie hiërarchie van Klassen en Properties. Onderstaande figuur laat zien dat er extra `rdfs:subClassOf` relaties afgeleid kunnen worden (rode pijl kan afgeleid worden).



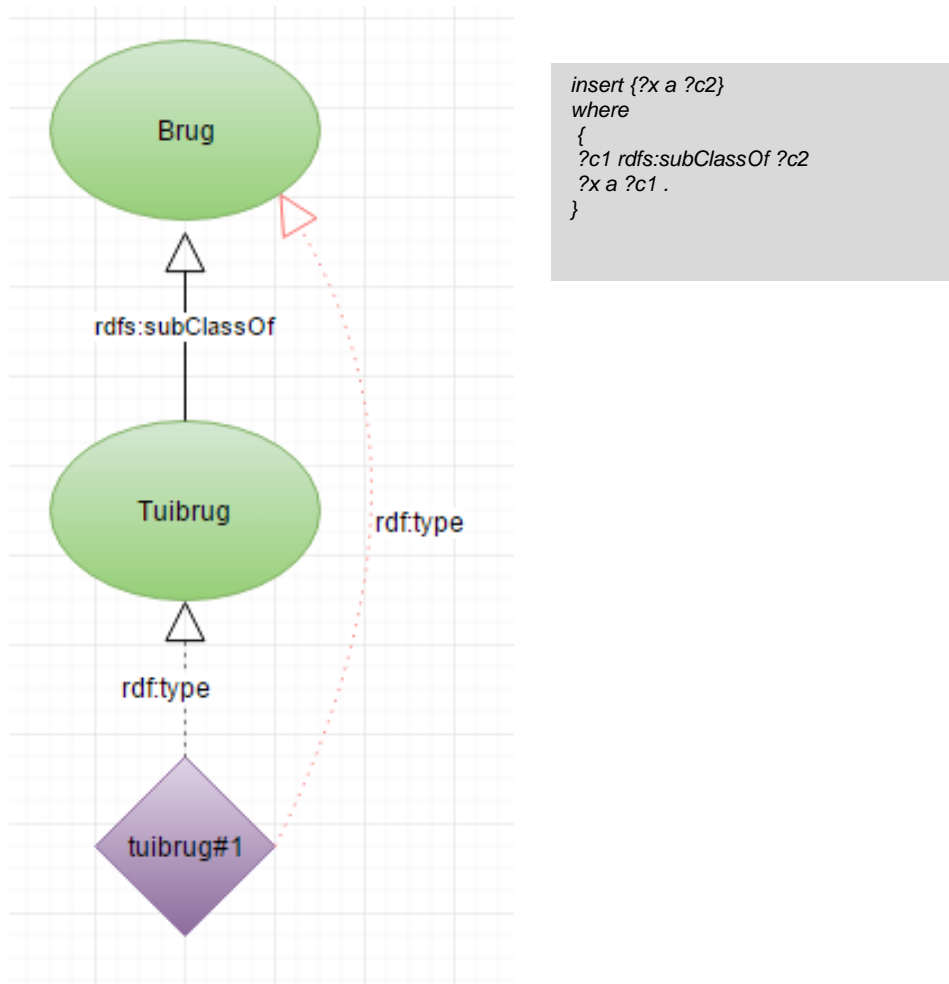
Om deze afleiding formeel te beschrijven kan de volgende SPARQL query gedefinieerd worden.

```
insert {?C rdfs:subClassOf ?A}
WHERE
{
    ?C rdfs:subClassOf ?B .
    ?B rdfs:subClassOf ?A .
    filter not exists{?C rdfs:subClassOf ?A}
}
```

Deze Sparql query definieert de precieze situatie waarbij de afleiding altijd gemaakt mag worden. Daarmee is een stukje semantiek van COINS 2.0 geformaliseerd.

3.3 Instance semantiek

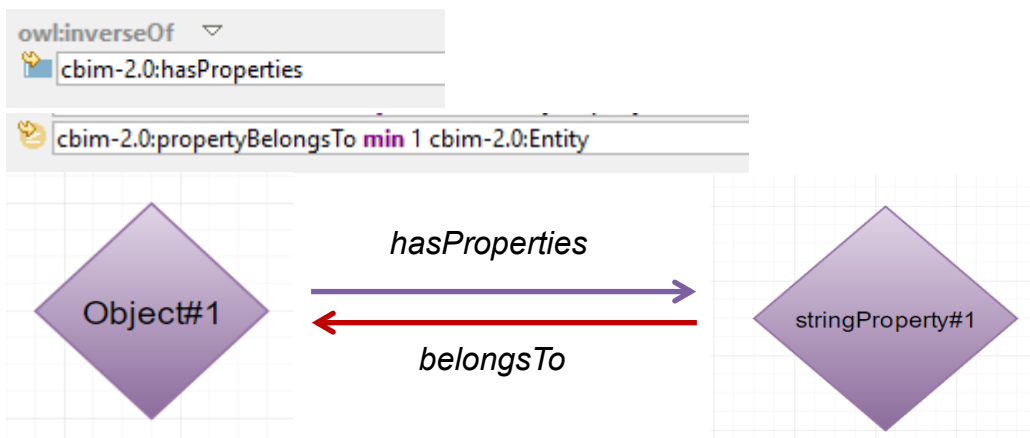
Ook op instantie nivo zijn afleidingen mogelijk. Onderstaande figuur laat zien dat er ‘typering’ afleidingen mogelijk zijn. Zo kan er een rdf:type relatie afgeleid worden voor ‘super’ Klasse. Ook deze afleiding is weer heel goed te formaliseren via de volgende Sparql Insert query.



3.4 Afleidingen en validatie

In COINS 2.0 Core model wordt gebruik gemaakt van RDF, RDFS en OWL vocabulair zoals rdf:type, rdfs:subClassOf, etc. Afleidingen zijn noodzakelijk om de gehele dataset expliciet te maken. Wanneer deze expliciet is kan er gevalideerd worden. De noodzaak voor deze afleidingen wordt met het volgende voorbeeld gegeven.

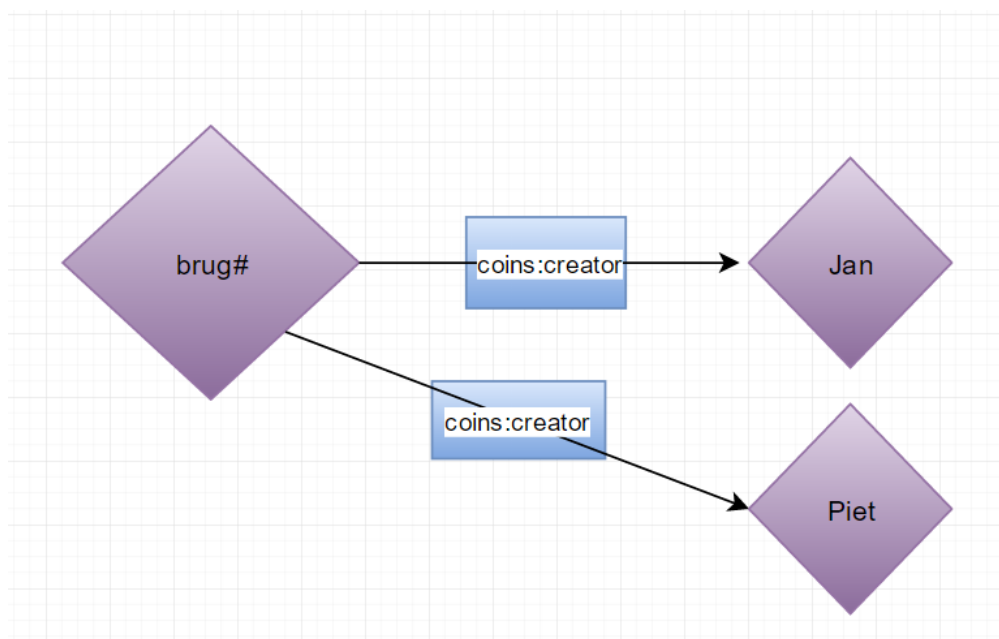
*In COINS 2.0 **moeten** EntityProperties altijd een ‘propertyBelongsTo’ relatie hebben. Dit is gedefinieerd via een owl:minCardinality constraint die COINS 2.0 interpreteert als een constraint waaraan voldaan moet worden. Zie onderstaande figuren waarin de ‘belongsTo’ relatie afgeleid kan worden waardoor er voldaan wordt aan de constraint.*



Ook de validatie op basis van de owl cardinality constraint is weer te formaliseren via een Sparql query.

3.5 Ongewenste OWA OWL afleidingen

Een belangrijke reden voor het formaliseren van de COINS 2.0 semantiek is het feit dat niet alle gestandaardiseerde afleidingen volgens de Open World Aannamen (OWA) van OWL wenselijk zijn voor COINS 2.0. Zo is bijvoorbeeld de COINS property 'creator' functional verklaard. COINS 2.0 ziet onderstaande dataset als invalide terwijl OWL extra afleidingen kan maken. In onderstaande figuur zou bijvoorbeeld de ongewenste conclusie getrokken kunnen worden dat 'Jan' gelijk is aan 'Piet'.



3.6 Conclusies

Het RDFS/OWL vocabulair maakt het mogelijk om extra informatie af te leiden. Voor validatie van COINS "data" sets is het noodzakelijk om de dataset expliciet te maken. Dit geldt voor 'schema' ontologieën maar ook voor "data" ontologieën.

Sparql queries kunnen gebruikt worden om de mogelijke afleidingen eenduidig te formaliseren met directe executie/implementatie mogelijkheden. Sparql queries kunnen tevens gebruikt worden om de 'data' validatie te formaliseren met directe executie/implementatie mogelijkheden.

4 Coins 2.0 Lite en COINS 2.0 Lite+eq profielen

4.1 2 profielen

COINS 2.0 is gebaseerd op OWL/RDF waardoor afleiden van extra feiten mogelijk is. Dit hoofdstuk definieert 2 profielen te weten COINS 2.0 Lite en COINS 2.0 Lite+eq. Deze profielen definiëren de preciese betekenis die COINS 2.0 hanteert mbt het OWL vocabulair. De profielen formaliseren alleen de betekenis voor een subset van het gehele OWL vocabulair. Toekomstige profielen kunnen wellicht meer OWL vocabulair formaliseren. Dit hoofdstuk definieert 2 profielen te weten COINS 2.0 Lite en COINS 2.0 Lite+Eq. Het eerste profiel bevat een bruikbare subset van OWL vocabulair waarmee afleidingen en validaties relatief snel uitgevoerd kunnen worden. Het COINS 2.0 Lite+eq profiel voegt OWL 'equivalent' class mogelijkheden toe aan het COINS 2.0 Lite profie is daarmee een 'zwaarder' profiel dat meer verwerkings capaciteit vraagt.

4.2 COINS semantiek overzicht

4.2.1 Cardinaliteits restricties

De COINS semantiek interpreteert de OWL cardinaliteits restricties als vocabulair om datasets te kunnen valideren volgens CWA. Dat wil zeggen dat cardinaliteits restricties gebruikt worden om datasets te testen. Indien data niet aanwezig is terwijl deze wel verwacht wordt volgens cardinaliteits restricties wordt een dataset als invalide beschouwd.

4.2.2 Functionalproperty

Een functional property wordt geïnterpreteerd als een CWA restrictie waarmee datasets getoetst kunnen worden. Meerdere unieke waarden voor 1 functional property is niet toegestaan en daarmee is een dataset invalide.

4.2.3 Domain en Range van owl:properties

Domain en Range properties van owl:properties worden geïnterpreteerd als data validatie regels op basis van CWA. Daarmee worden datasets gevalideerd of relaties wel tussen de juiste objecten zijn gelegd. Het is dus noodzakelijk om individuals expliciet te typeren.

4.2.4 Union class

Individuals van Classes gedefinieerd als subclasses van Union classes worden als abstract beschouwd in COINS. Deze individuals vereisen dus verdere typering.

4.2.5 ExpiredEntity

ExpiredEntities doen niet mee in cardinaliteits restricties. Dat wil zeggen dat individuals lid van ExpiredEntities niet meetellen bij cardinaliteits controles van COINS Containers.

4.3 COINS Lite vocabulair

4.3.1 COINS 2.0 Lite vocabulair

Het COINS Lite vocabulair is gebaseerd op RDF/RDFS en ondersteunt een groot gedeelte van de OWL vocabulair met een Closed World interpretatie. Hieronder staat een lijst met OWL classes waarvan de betekenis(semantiek) is geformaliseerd in de COINS 2.0 lite profielen. De daadwerkelijke formalisering hiervan wordt in het volgende hoofdstuk beschreven.

#	Owl classes
---	-------------

1	owl:AllDisjointClasses
2	owl:Annotation
3	owl:Thing
4	owl:TransitiveProperty
5	owl:Restriction
6	owl:Ontology
7	owl:ObjectProperty
8	owl:NamedIndividual
9	owl:FunctionalProperty
10	owl:DatatypeProperty
11	owl:Class
12	owl:AnnotationProperty

De volgende lijst geeft de OWL predicaten aan waarvan semantiek is geformaliseerd voor COINS 2.0 in de COINS lite profielen

#	Owl Predicates
1	owl:allValuesFrom
2	owl:unionOf
3	owl:someValuesFrom
4	owl:qualifiedCardinality
5	owl:onProperty
6	owl:oneOf
7	owl:onClass
8	owl:minQualifiedCardinality
9	owl:minCardinality
10	owl:members
11	owl:maxQualifiedCardinality
12	owl:maxCardinality
13	owl:inverseOf
14	owl:intersectionOf
15	owl:imports
16	owl:hasValue
17	owl:distinctMembers
18	owl:disjointWith
19	owl:cardinality

De volgende lijst bevat OWL vocabulair zonder formele betekenis. Dit vocabulair is toegestaan in de lite profielen maar hebben geen speciale betekenis.

#	
1	owl:annotatedProperty
2	owl:annotatedSource
3	owl:annotatedTarget
4	owl:deprecated
5	owl:priorVersion
6	owl:versionInfo
7	owl:versionIRI
8	owl:incompatibleWith
9	owl:backwardCompatibleWith

4.3.2 COINS 2.0 Lite+Eq vocabulair

Het COINS 2.0 Lite+Eq profiel is een uitbreiding van het COINS 2.0 Lite profiel en formaliseert daarnaast nog de betekenis van het predicaat owl:equivalentClass. De volgende regel wordt dus toegevoegd aan de OWL predicaten.

20	owl:equivalentClass
----	---------------------

4.4 COINS Referentie kader en COINS OTL profiel testen

Om van te voren te weten of een uitbreiding van COINS zoals een referentie kader of OTL binnen een profiel valt zijn er 2 SPARQL queries opgesteld die dit kunnen toetsen:

- 1) COINS 2.0 Lite toets
- 2) COINS 2.0 Lite+Eq toets

4.4.1 COINS 2.0 Lite profiel test

De volgende Sparql queries detecteren of een referentie kader/OTL binnen de OTL Lite profiel valt. Indien er geen resultaten gevonden worden valt de uitbreiding binnen dit profiel. Deze queries kunnen zonder extra imports of afleidingen gebruikt worden.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?a ?b ?c

WHERE {Graph <referentiekader of OTL>{
  ?a ?b ?c filter (STRSTARTS(STR(?b),"http://www.w3.org/2002/07/owl#")) .filter (?b
not in
(owl:allValuesFrom,owl:cardinality,owl:disjointWith,owl:distinctMembers,owl:hasValue,ow
l:imports,owl:intersectionOf,owl:inverseOf,owl:maxCardinality,owl:maxQualifiedCardinali
ty,owl:members,owl:minCardinality,owl:minQualifiedCardinality,owl:onClass,owl:oneOf,owl
:onProperty,owl:qualifiedCardinality,owl:someValuesFrom,owl:unionOf,owl:equivalentClass
,owl:annotatedProperty,owl:annotatedSource,owl:annotatedTarget,owl:backwardCompatibleWi
th,owl:deprecated,owl:incompatibleWith,owl:priorVersion,owl:versionInfo,owl:versionIRI)
)}}}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?a ?b ?c

WHERE {Graph < referentiekader of OTL >{
  ?a ?b ?c filter (STRSTARTS(STR(?c),"http://www.w3.org/2002/07/owl#")) .filter (?c
not in
(owl:AllDisjointClasses,owl:Annotation,owl:AnnotationProperty,owl:Class,owl:DatatypePro
perty,owl:FunctionalProperty,owl:NamedIndividual,owl:ObjectProperty,owl:Ontology,owl:Re
striction,owl:TransitiveProperty,owl:Thing))
}}}
```

4.4.2 COINS lite+eq profil test

De volgende Sparql queries detecteren of een referentie kader/OTL binnen de OTL Lite+Eq profil valt. Indien er geen resultaten gevonden worden valt de uitbreiding binnen dit profil. Deze queries kunnen zonder extra imports of afleidingen gebruikt worden.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?a ?b ?c

WHERE {Graph <referentiekader of OTL >{
  ?a ?b ?c filter (STRSTARTS(STR(?c),"http://www.w3.org/2002/07/owl#")) .filter (?c
not in
(owl:AllDisjointClasses,owl:Annotation,owl:AnnotationProperty,owl:Class,owl:DatatypePro
perty,owl:FunctionalProperty,owl:NamedIndividual,owl:ObjectProperty,owl:Ontology,owl:Re
striction,owl:TransitiveProperty,owl:Thing)
}}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?a ?b ?c

WHERE {Graph <referentiekader of OTL>{
  ?a ?b ?c filter (STRSTARTS(STR(?b),"http://www.w3.org/2002/07/owl#")) .filter (?b
not in
(owl:allValuesFrom,owl:cardinality,owl:disjointWith,owl:distinctMembers,owl:hasValue,ow
l:imports,owl:intersectionOf,owl:inverseOf,owl:maxCardinality,owl:maxQualifiedCardinali
ty,owl:members,owl:minCardinality,owl:minQualifiedCardinality,owl:onClass,owl:oneOf,owl
:onProperty,owl:qualifiedCardinality,owl:someValuesFrom,owl:unionOf,owl:equivalentClass
,owl:annotatedProperty,owl:annotatedSource,owl:annotatedTarget,owl:backwardCompatibleWi
th,owl:deprecated,owl:incompatibleWith,owl:priorVersion,owl:versionInfo,owl:versionIRI)
}
}
```

5 COINS 2.0 Lite en Lite+Eq Semantiek

5.1 Inleiding

Dit hoofdstuk beschrijft de COINS 2.0 Lite en COINS 2.0 Lite+eq semantiek. De semantiek wordt geformaliseerd via Sparql Select en Sparql Insert queries. De Sparql Insert queries zijn executeerbaar en voegen afleidingen toe aan de database. Met de Sparql Insert queries zijn de afleidingen op basis van RDF/RDFS/OWL vocabulair geformaliseerd. De Sparql Select queries worden gebruikt voor validatie van de semantisch BIM dataset. Indien deze Sparql Select queries resultaten vinden correspondeert dit met onregelmatigheden in de dataset.

Het doel van deze queries is om de semantiek van COINS 2.0 te definiëren. Bij implementatie kan er natuurlijk afgeweken worden van deze queries zolang het resultaat hetzelfde is.

De volgende paragrafen geven een overzicht van alle sparql queries die gezamenlijk de COINS 2.0 Lite(+EQ) semantiek formaliseren. De daadwerkelijke Sparql queries zijn te vinden in de bijlage.

5.2 Schema afleidingen

Op 'schema' nivo zoals het COINS 2.0 model, OTL's en Referentie kaders zijn de volgende Sparql queries gedefinieerd. Deze queries leiden nieuwe informatie af die toegevoegd dient te worden aan de 'schema' dataset opdat weer nieuwe afleidingen evt gemaakt kunnen worden. Dit wil zeggen dat deze queries als een "Rule" geïmplementeerd dienen te worden.

#	Sparql	Beschrijving	Lite	Lite+ Eq	Full
49	scm-avf1	Subclass afleiding tussen restrictions	x		
50	scm-avf1	Subclass afleiding tussen restrictions (andersom)	x		
51	Scm-cls	Standard owl toevoegingen voor classes (subclass van zichzelf, equivalent aan zichzelf, subclass nothing)			x
52	Scm-dom1	Domain subtype toevoegingen voor relaties	x		
53	Scm-dom2	Domain toevoeging voor subproperties	x		
54	Scm-dp	Standaard datatype property toevoegingen(subproperty van zichzelf, equivalent van zichzelf)			x
55	Scm-eqc1	Subclassing toevoeging van equivalent classes	x		
56	Scm-eqc2	Subclassing toevoeging van equivalent classes andersom	x		
57	Scm-eqp1	subProperty toevoegingen opbasis van equivalentProperties		x	
58	Scm-eqp1	subProperty toevoegingen opbasis van equivalentProperties		x	
59	Scm-hv	Subclass toevoeging van restricties	x		
60	Scm-int	Subclass toevoeging op basis van intersectionOf	x		
61	Scm-op	Standaard objectproperty toevoegingen		x	
62	Scm-rng1	Range toevoegingen voor subProperties	X		
63	Scm-rng2	Range toevoegingen voor subProperties	X		
64	Scm-sco	Subclass hierarchy toevoegen	X		
65	Scm-spo	Subproperty hierarchy toevoegen	X		
66	Scm-svf1	someValues of restriction subtyping toevoegen	x		
67	Scm-svf2	someValues of restriction subtyping toevoegen	x		
68	Scm-uni	Subclass toevoeging op basis van unionOf	x		

5.3 Data afleidingen

De volgende tabel geeft een overzicht van alle Sparql queries die extra data afleiden voor het semantisch BIM model. De afleidingen moeten toegevoegd worden aan de dataset opdat er weer meer afleidingen gemaakt kunnen worden. Dit wil zeggen dat deze queries als een “Rule” geïmplementeerd dienen te worden.

#	Sparql	Beschrijving	Lite	Lite+Eq	Full	NIET
16	Cax-eqc1	equivalentClass typering van individuals		x		
17	Cax-eqc2	Equivalent class typering van individuals		x		
18	Cax-sco	Subclass typering toevoegen aan individuals	x			
19	Cls-avf	Typering van propertyValue volgens allValuesFrom constraint. Dit willen we niet				x
20	Cls-hv1	Automatisch hasValue waardes toevoegen	x			
21	Cls-hv2	Automatisch typering volgens hasValue on property		X		
22	Cls-int1	Intersectie typering toevoegen vanaf (non-intersectie klasse)		X		
23	Cls-int2	Intersectie typering toevoegen vanaf intersectie klasse				x
24	Cls-maxc2	Individual typering op basis van 2 propertyvalues met een max =1.				x
25	Cls-maxqc3	Individual 1 =individual 2 op basis cqr max 1.				x
26	Cls-maxqc4	Individual 1 =individual 2 op basis cqr max 1.				x
27	Cls-oo	Owl:one of typering van individuals	x			
28	Cls-svf1	Individual typering op basis van property someValues of		X		
29	Cls-svf2	Individual typering op basis van property someValues of		x		
30	Cls-uni	Individual typering naar een union class	x			
31	Eq-rep-o	Kopieren eigenschappen van sameAs individuals (objects)			x	
32	Eq-rep-p	Kopieren eigenschappen van sameAs individuals (predicates)			x	
33	Eq-rep-s	Kopieren eigenschappen van sameAs individuals (subjects)			x	
34	Eq-sym	Y=x dan x=y (owl:sameAs)			x	
35	Eq-trans	x=y en y=z dan x=z (owl:sameAs)			x	
36	Prp-dom	Domain class gebruiken om individuals te typeren.				x
37	Prp-eqp1	Kopieer alles van equivalent properties			x	
38	Prp-eqp2	Kopieer alles van equivalent properties (andersom)			x	
39	Prp-fp	propertyValueA=propertyValueB bij functionele properties.				x
40	Prp-ifp	propertyValueA=propertyValueB bij inverse functionele properties.				x
41	Prp-inv	Inverse property values toevoegen	x			

42	Prp-inv2	Inverse property values toevoegen (andersom)	x	
43	Prp-key	Haskey		x
44	Prp-rng	Individual typing via Range construct.		x
45	Prp-spo1	Toevoeging van super properties voor individuals	x	
46	Prp-spo2	Propertychain value		x
47	Prp-symp	Symmetrische properties toevoeging		x
48	Prp-trp	Transitive property toevoegingen	x	
49	scm-avf1	Subclass afleiding tussen restrictions	x	
50	scm-avf1	Subclass afleiding tussen restrictions (andersom)	x	
51	allValuesFrom typing	allValuesFrom typing		x

5.4 Validatie

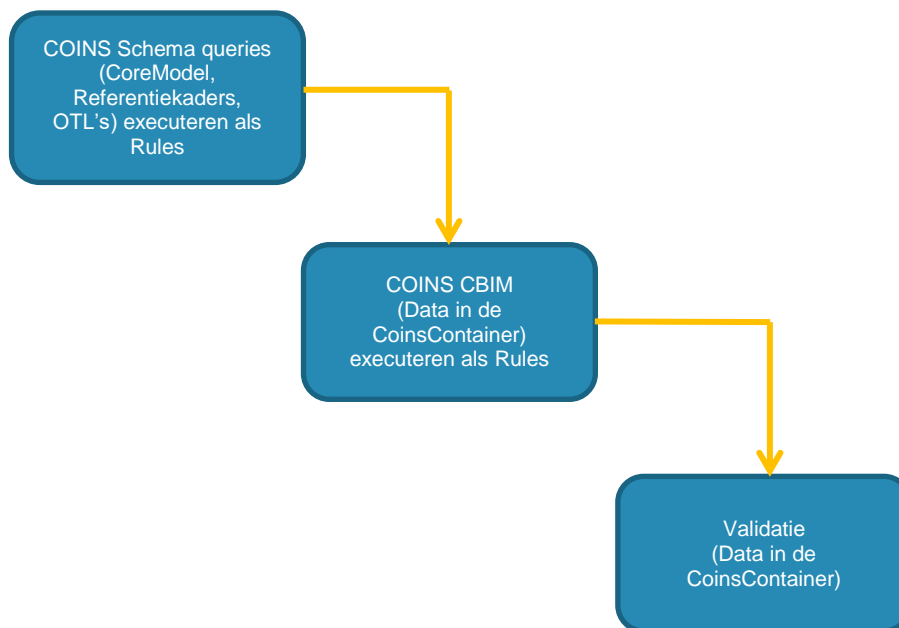
De volgende tabel geeft een overzicht van alle validatie queries. Indien deze queries resultaten geven wil dat zeggen dat een bepaalde 'constraint' overtreden is .

#	Sparql	Beschrijving
1	Cax-adx	Zijn er individuals lid van klassen die disjoint zijn verklaard via de AllDisjointClasses vocabulair
2	Cax-dw	Zijn er individuals lid van klassen die disjoint zijn verklaard via de owl:disjointwith vocabulair
3	Cls-maxc1	Wordt een max cardinaliteit overschreden
4	Cls-maxqc1	Wordt er een max cardinaliteit overschreden via een QCR met een juiste klasse
5	Cls-maxqc2	Wordt er een max cardinaliteit overschreden via een QCR
6	COINS-CCO	Alleen individuals van COINS-Concept class
7	COINS-NSWO	Geen OWL Class definities
8	COINS-NSCO	Geen (Rdf) subclass definities
9	COINS-NSPO	Geen (Rdf) subProperties definities
10	COINS-OPVU	IRI waarde check voor ObjectProperties
11	COINS-DPVL	Literal waarde check voor DatatypeProperties
12	COINS-DTVC	Datatype check voor Literal values van DatatypeProperties
13	COINS-minCar	Minimum cardinality constraint validatie
14	COINS-maxCar	Maximum cardinality constraint validatie
15	COINS-FUP	FunctionalProperty als een restrictie validation
16	COINS-Carex	Exactly Cardinality constraint validatie
17	COINS-QCREx	Qualitative Cardinality Restriction Exactly validatie
18	COINS-QCRMin	Qualitative Cardinality Restriction Minimum validatie
19	COINS-QCRMax	Qualitative Cardinality Restriction Maximum validatie
20	COINS-UO	UnionOf als een Restrictie validatie
21	COINS-IS	Intersection als een Restrictie validatie
22	COINS-dom	Domain als een restrictie
23	COINS-rng	Range als een restrictie

5.5 Executie van de Sparql queries

Afleidingen zorgen voor nieuwe informatie waardoor er wellicht weer nieuwe afleidingen nodig zijn. De Sparql queries die voor nieuwe informatie zorgen moeten zich gedragen als een “rule” waardoor dus ook gebruik gemaakt wordt van nieuwe informatie. Wel is het mogelijk om eerst de Schema afleidingen door te voeren en daarna de Data afleidingen. Deze moeten dus beide als rule geïmplementeerd worden.

Na deze afleidingen is de dataset expliciet en kan de dataset gevalideerd worden met de validatie sparql queries. Deze geven resultaten indien er fouten gevonden zijn in de dataset. Deze validatie queries voegen geen nieuwe informatie toe aan de dataset. Onderstaande schema geeft de workflow aan van de sparql queries.



Deze situatie is geldig wanneer er geen ‘data’ in de schema ontologien zit. Zit er wel ‘data’ in de schema ontologien dan moet deze dataset ook expliciet gemaakt worden via de ‘data’ queries.

6 Naar een COINS 2.0 validatie service & tool

6.1 Prototype implementatie

Binnen dit project is een Java prototype gemaakt waarin alle queries op een dataset gerund kunnen worden. De sparql queries zijn “geparameteriseerd” waardoor Graphnamen snel veranderd kunnen worden. Ook is er een simpele Java Interface gemaakt om tot sparql executie te komen. Indien deze Interface geïmplementeerd wordt kan er gevalideerd worden. Deze code is overdraagbaar voor een definitieve validatie service.

6.2 COINS 2.0 API uitbreiding

De sparql queries zoals beschreven in dit document kunnen makkelijk gekoppeld worden aan de COINS 2.0 API. De Api is namelijk al klaar voor Sparql queries. Zo zouden alle sparql queries zoals beschreven in dit document gerund kunnen worden via de COINS 2.0 API. Rapportage van de fouten kan verder uitgewerkt worden. Ook is het mogelijk om de implementatie van het prototype te gebruiken.

6.3 Performance

Performance is helaas iets waar we (voorlopig) nog rekening mee moeten houden. Er zijn meerdere optimalisaties mogelijk zoals:

- 1) Goede Triplestore met hoge (Sparql) performance
- 2) Klaar zetten van Schema's opdat deze niet meer door het afleiden/validatie proces moeten
- 3) Gebruik maken van RDFS 'built-in' routines indien aanwezig
- 4) Query rewrite / Sparql-DL voor minder afleidingen en meer kennis in de queries
 - a. Zeker een rewrite van de queries opdat niet alle rdf:type materialisatie meer nodig is kan wellicht kansrijk zijn met wellicht significante voordelen.
- 5) Goede hardware en evt clusterings principes gebruiken
- 6) Omzetten van de Sparql queries naar (native) Triplestore rules

6.4 Aanvullende validaties voor een COINS 2.0 Validatie service

De volgende **extra** validaties zijn handig voor een COINS 2.0 validatie service:

- 1) Zip file validatie
 - a. Valide zip file
 - b. Benodigde directory structuur aanwezig en geen andere directories
- 2) WOA validatie
- 3) Document validatie
 - a. Zijn alle documenten in de doc folder ook gebruikt in het semantisch BIM?
 - b. Zijn alle documenten in het semantisch BIM ook aanwezig in de doc folder

6.5 Extra eisen voor een Validatie Service

De volgende eisen zijn alvast verzameld voor een validatie service:

- 1) Uploaden van een (subset) van een COINS container
- 2) Profielkeuze mogelijkheid (Lite en Lite+EQ en wellicht meer)
- 3) Admin gedeelte voor het aanmaken/toevoegen van nieuwe profielen

- 4) Validatie API opdat deze service een output kan genereren voor andere services en dus mee kan doen in een SOA omgeving

7 Bijlage 1: Schema Entailment Sparql queries

#name :subclassing allvaluesfrom restrictions part 1
#description :Entailment of restriction subclasses via the allvaluesfrom construct part 1
#reference :scm-avf1 (49)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{  
  ?c1 rdfs:subClassOf ?c2  
}}
```

```
WHERE {Graph <schemaGraph>{  
  ?c1 owl:allValuesFrom ?y1 .  
  ?y1 rdfs:subClassOf ?y2. ?c2 owl:allValuesFrom ?y2 .  
  ?c1 owl:onProperty ?p .  
  ?c2 owl:onProperty ?p  
}  
}
```

#name :subclassing allvaluesfrom restriction part 2
#description :Entailment of restriction subclasses via the allvaluesfrom construct part 2
#reference :scm-avf2 (50)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{  
  ?c2 rdfs:subClassOf ?c1  
}}
```

```
WHERE {Graph <schemaGraph>{  
  ?c1 owl:allValuesFrom ?y . ?c1 owl:onProperty ?p1 . ?c2 owl:allValuesFrom ?y . ?c2 owl:onProperty ?p2 . ?p1 rdfs:subPropertyOf  
  ?p2  
}  
}
```

#name :Domain additions for subproperties
#description :entailment of domains for subproperties
#reference :scm-dom2 (53)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{  
  ?p1 rdfs:domain ?c  
}}
```

```
WHERE {Graph <schemaGraph>{  
  ?p2 rdfs:domain ?c . ?p1 rdfs:subPropertyOf ?p2  
}  
}
```

#name :Restriction subclassing for hasValue restrictions
 #description :Entailment of restriction subclasses via the hasValue construct
 #reference :scm-hv (59)
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{
  ?c1 rdfs:subClassOf ?c2
}}
```

```
WHERE {Graph <schemaGraph>{
  ?c1 owl:hasValue ?i . ?c1 owl:onProperty ?p1 . ?c2 owl:hasValue ?i . ?c2 owl:onProperty ?p2 . ?p1 rdfs:subPropertyOf ?p2
}
}
```

#name :Subclass based upon intersection classes
 #description :Restriction subclassing for hasValue restrictions. dependency on list:member removed
 #reference :scm-int (60) (member:list replaced)
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{
  ?c rdfs:subClassOf ?cl
}}
```

```
WHERE {Graph <schemaGraph>{
  ?c owl:intersectionOf /rdf:rest*/rdf:first ?cl
}
}
```

#name :Scm-rng1 (62)
 #description :entailment of ranges for subclasses
 #reference :Scm-rng1 (62)
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{
  ?p rdfs:range ?c2
}}
```

```
WHERE {Graph <schemaGraph>{
  ?p rdfs:range ?c1 . ?c1 rdfs:subClassOf ?c2
}
}
```

#name :Scm-rng2 (63)
 #description :entailment of ranges for subproperties

#reference :Scm-rng2 (63)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{  
  ?p1 rdfs:range ?c  
}}
```

```
WHERE {Graph <schemaGraph>{  
  ?p2 rdfs:range ?c .    ?p1 rdfs:subPropertyOf ?p2  
}  
}
```

#name :Scm-sco (64)
#description :entailment of the subclass hierarchy
#reference :Scm-sco (64)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{  
  ?c1 rdfs:subClassOf ?c3  
}}
```

```
WHERE {Graph <schemaGraph>{  
  ?c1 rdfs:subClassOf ?c2 .    ?c2 rdfs:subClassOf ?c3  
}  
}
```

#name :Scm-spo (65)
#description :entailment of the subproperty hierarchy
#reference :Scm-spo (65)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
insert {Graph <schemaGraph>{  
  ?p1 rdfs:subPropertyOf ?p3  
}}
```

```
WHERE {Graph <schemaGraph>{  
  ?p1 rdfs:subPropertyOf ?p2 .    ?p2 rdfs:subPropertyOf ?p3  
}  
}
```

#name :scm-svf1 (66)
#description :Restriction subclassing based upon someValuesFrom
#reference :scm-svf1 (66)
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <schemaGraph>{
  ?c1 rdfs:subClassOf ?c2
}}
```

```
WHERE {Graph <schemaGraph>{
  ?c1 owl:someValuesFrom ?y1 . ?c1 owl:onProperty ?p . ?c2 owl:someValuesFrom ?y2 . ?c2 owl:onProperty ?p . ?y1 rdfs:subClassOf
  ?y2
}
}
```

#name	:scm-svf2 (67)
#description	:Restriction subclassing based upon someValuesFrom
#reference	:scm-svf2 (67)
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <schemaGraph>{
  ?c1 rdfs:subClassOf ?c2
}}
```

```
WHERE {Graph <schemaGraph>{
  ?c1 owl:someValuesFrom ?y . ?c1 owl:onProperty ?p1 . ?c2 owl:someValuesFrom ?y . ?c2 owl:onProperty ?p2 . ?p1
  rdfs:subPropertyOf ?p2
}
}
```


8 Bijlage 2: Instance Entailment Sparql queries

#name	:Subclass typing
#description	:Entailment of superclass typing
#reference	:CAXSCO (18)
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <instanceGraph>{
  ?x a ?c2
}}
```

```
WHERE {Graph <instanceGraph>{
  ?x a ?c1
}
Graph <schemaGraph>{
  ?c1 rdfs:subClassOf ?c2
}
}
```

#name	:Subproperty value entailment
#description	:Entailment of subproperty values
#reference	:Prp-spo1 (45)
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <instanceGraph>{
  ?x ?p2 ?y
}}
```

```
WHERE {Graph <schemaGraph>{
  ?p1 rdfs:subPropertyOf ?p2
}
Graph <instanceGraph>{
  ?x ?p1 ?y .
}
}
```

#name	:Transitive property values
#description	:Entailment of transitive property values
#reference	:Prp-trp (48)
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <instanceGraph>{
  ?x ?p ?z
}}
```

```
WHERE {Graph <schemaGraph>{
  ?p owl:TransitiveProperty
}
Graph <instanceGraph>{
  ?x ?p ?y .          ?y ?p ?z
}
```

```
}
}
```

```
#name          :Default values via owl:hasValue
#description    :Default values entailment via the hasValue constraint
#reference      :cls-hv1 (20)
#profile        :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <instanceGraph>{
?u ?p ?y
}}
```

```
WHERE {Graph <schemaGraph>{
?x owl:hasValue ?y . ?x owl:onProperty ?p .
}
Graph <instanceGraph>{
?u a ?x .
}
}
```

```
#name          :InverseOf value entailment part 2
#description    :Entailment of inverseOf values part 1
#reference      :Prp-inv (41)
#profile        :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <instanceGraph>{
?y ?p2 ?x
}}
```

```
WHERE {Graph <schemaGraph>{
?p1 owl:inverseOf ?p2 .
}
Graph <instanceGraph>{
?x ?p1 ?y .
}
}
```

```
#name          :InverseOf value entailment part 2
#description    :Entailment of inverse values part 2
#reference      :Prp-inv2 (42)
#profile        :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
insert {Graph <instanceGraph>{
?y ?p1 ?x
}}
```

```
WHERE {Graph <schemaGraph>{
?p1 owl:inverseOf ?p2
}
Graph <instanceGraph>{
```

```

    ?x ?p2 ?y .
}
}

```

```

#name          :UnionOf typing
#description    :Typing to Union Class entailment
#reference      :Cls-uni (30)
#profile       :COINS 2.0 Lite

```

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```

insert {Graph <instanceGraph>{
  ?y a ?c
}}

```

```

WHERE {Graph <instanceGraph>{
  ?y a ?ci
}
Graph <schemaGraph>{
  ?c owl:unionOf ?x . ?x rdf:rest*/rdf:first ?ci
}
}

```

9 Bijlage 3: Data Validation Sparql queries

#name	:Minimal Cardinality validatie
#description	:minimal cardinality constraint validation
#reference	:COINS-minCar
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?this ?min ?aantal ?property

WHERE {{
select distinct ?this ?class ?min (count(distinct ?value) as ?aantal) ?property

WHERE {Graph <schemaGraph>{
  ?class a owl:Restriction . ?class owl:minCardinality ?min . ?class owl:onProperty ?property
}
Graph <instanceGraph>{
  {?this a ?class . ?this ?property ?value. filter not exists {?value a <http://www.coinsweb.nl/cbim-2.0.rdf#ExpiredEntity>}} union
  {?this a ?class . filter not exists {?this ?property ?value}}
}

}group by ?this ?class ?min ?property
}
filter(?aantal<?min)

}
```

#name	:Cardinality 'exactly' validatie
#description	:Exactly cardinality validatie
#reference	:COINS-CAREX
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?this ?min ?aantal ?property

WHERE {{
select distinct ?this ?min (count(distinct ?value) as ?aantal) ?property

WHERE {Graph <schemaGraph>{
  ?class a owl:Restriction . ?class owl:cardinality ?min . ?class owl:onProperty ?property .
}
Graph <instanceGraph>{
  {?this a ?class . ?this ?property ?value. filter not exists {?value a <http://www.coinsweb.nl/cbim-2.0.rdf#ExpiredEntity>}} union
  {?this a ?class . filter not exists {?this ?property ?value}}
}

} group by ?class ?this ?min ?property
}
filter((?aantal<?min) || (?aantal>?min) )

}
```

#name	:Maximum Cardinality validatie
#description	:maximum cardinality constraint validation
#reference	:COINS-maxCar
#profile	:COINS 2.0 Lite

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```

select distinct ?this ?class ?min ?aantal ?property

WHERE {{
select distinct ?this ?min (count(distinct ?value) as ?aantal) ?property

WHERE {Graph <schemaGraph>{
?class a owl:Restriction . ?class owl:maxCardinality ?min . ?class owl:onProperty ?property
}
Graph <instanceGraph>{
?this a ?class . ?this ?property ?value. filter not exists {?value a <http://www.coinsweb.nl/cbim-2.0.rdf#ExpiredEntity>}
}

}group by ?this ?class ?min ?property
}
filter(?aantal>?min)

}

#name :Qualitative Cardinality Restriction 'exactly' validatie
#description :Exactly Qualitative Cardinality restriction validatie
#reference :COINS-QCREx
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?this ?min ?aantal ?property

WHERE {{
select distinct ?this ?class ?min (count(distinct ?value) as ?aantal) ?property

WHERE {Graph <schemaGraph>{
?class a owl:Restriction . ?class owl:onClass ?onClass . ?class owl:qualifiedCardinality ?min . ?class owl:onProperty ?property .
}
Graph <instanceGraph>{
?this a ?class . optional{ ?this ?property ?value. filter not exists {?value a
<http://www.coinsweb.nl/cbim-2.0.rdf#ExpiredEntity>}. ?value a ?onClass }
}

} group by ?this ?class ?min ?property
}
filter((?aantal<?min) || (?aantal>?min) )

}

#name :Qualitative Cardinality Restriction maximum validatie
#description :Maximum Qualitative Cardinality restriction validatie
#reference :COINS-QCRMax
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?this ?class ?min ?aantal ?property

WHERE {{
select distinct ?this ?class ?min (count(distinct ?value) as ?aantal) ?property

WHERE {Graph <schemaGraph>{
?class a owl:Restriction . ?class owl:onClass ?onClass . ?class owl:maxQualifiedCardinality ?min . ?class owl:onProperty
?property .
}
Graph <instanceGraph>{
?this a ?class . optional{ ?this ?property ?value. filter not exists {?value a <http://www.coinsweb.nl/cbim-
2.0.rdf#ExpiredEntity>}.?value a ?onClass}
}

}

```

```

} group by ?this ?class ?min ?property
}
filter((?aantal>?min) )
}

```

```

#name :Qualitative Cardinality Restriction Minimum validatie
#description :Minimal Qualitative Cardinality restriction validatie
#reference :COINS-QCRMin
#profile :COINS 2.0 Lite

```

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```
select distinct ?this ?min ?aantal ?property
```

```

WHERE {{
select distinct ?this ?class ?min (count(distinct ?value) as ?aantal) ?property

```

```

WHERE {Graph <schemaGraph>{
?class a owl:Restriction . ?class owl:onClass ?onClass. ?class owl:minQualifiedCardinality ?min . ?class owl:onProperty ?property
.
}
Graph <instanceGraph>{
?this a ?class . optional{ ?this ?property ?value. filter not exists {?value a
<http://www.coinsweb.nl/cbim-2.0.rdf#ExpiredEntity>}.?value a ?onClass}
}

```

```

} group by ?this ?class ?min ?property
}
filter((?aantal<?min) )
}

```

```

#name :FunctionalProperty aantal validatie
#description :functional property as a restriction validation
#reference :COINS-FUP
#profile :COINS 2.0 Lite

```

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```
select distinct ?this ?property ?aantal
```

```

WHERE {{
select distinct ?this ?property (count(distinct ?value) as ?aantal)

```

```

WHERE {Graph <schemaGraph>{
?property a owl:FunctionalProperty
}
Graph <instanceGraph>{
?this ?property ?value. filter not exists {?value a <http://www.coinsweb.nl/cbim-2.0.rdf#ExpiredEntity>}
}

} group by ?this ?property
}
filter(?aantal>1)
}

```

```

#name :Simple datatyping check
#description :Datatype check
#reference :COINS-DTVC

```

#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
select distinct ?prop ?range (Datatype(?value) as ?dat)

WHERE {Graph <schemaGraph>{
  ?prop a owl:DatatypeProperty .                ?prop rdfs:range ?range.
}
Graph <instanceGraph>{
  ?a ?prop ?value
}
filter (Datatype(?value) !=?range)

}
```

#name :Literal check for datatypes
#description :Datatype waarde check
#reference :COINS-DPVL
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
select distinct ?a ?prop ?value

WHERE {Graph <schemaGraph>{
  ?prop a owl:DatatypeProperty
}
Graph <instanceGraph>{
  ?a ?prop ?value. filter(!isLiteral(?value))
}

}
```

#name :ObjectProperty value type check
#description :ObjectProperty waarde check
#reference :COINS-OPVU
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
select distinct ?a ?prop ?value

WHERE {Graph <schemaGraph>{
  ?prop a owl:ObjectProperty
}
Graph <instanceGraph>{
  ?a ?prop ?value. filter(!isIRI(?value))
}

}
```

#name :domain als een restrictie validatie
#description :Domain als een restrictie
#reference :COINS-dom
#profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct *
```

```
WHERE {Graph <schemaGraph>{
  ?prop a owl:ObjectProperty. ?prop rdfs:domain ?domain.
}
Graph <instanceGraph>{
  ?a ?prop ?b
}
filter not exists {{Graph <instanceGraph>{
  ?a a ?domain
}} union { Graph <schemaGraph>{
  ?a a ?domain}
}}
}
```

```
#name :Range als restrictie validatie
#description :Range als een restrictie
#reference :COINS-rng
#profile :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct *
```

```
WHERE {Graph <schemaGraph>{
  ?prop a owl:ObjectProperty. ?prop rdfs:range ?range.
}
Graph <instanceGraph>{
  ?a ?prop ?b
}
filter not exists {{Graph <instanceGraph>{
  ?b a ?range
}} union { Graph <schemaGraph>{
  ?b a ?range}
}}
}
```

```
#name :UnionOf als een restrictie validatie
#description :UnionOf as a restriction validation
#reference :COINS-UO
#profile :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?restriction ?a ?first
```

```
WHERE {Graph <schemaGraph>{
  ?restriction owl:unionOf ?list.
}
Graph <instanceGraph>{
  ?a a ?restriction.
}
filter not exists{Graph <instanceGraph>{
  ?a a ?first.
}Graph <schemaGraph>{
  ?list rdf:rest*/rdf:first ?first
}}
}
```


#name :IntersectionOf
 #description :Intersection as a restriction validation
 #reference :COINS-IS
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?restriction ?a

```
WHERE {Graph <schemaGraph>{
  ?restriction owl:intersectionOf ?list.
}
Graph <instanceGraph>{
  ?a a ?restriction.
}
filter not exists{Graph <instanceGraph>{
  ?a a ?first
}Graph <schemaGraph>{
  ?list rdf:rest*/rdf:first ?first
}}
}
```

#name :validatie van alle objecten of ze minimaal een COINS Entity zijn
 #description :Only COINS concept instances check
 #reference :COINS-CCO
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?x ?type

```
WHERE {Graph <instanceGraph>{
  ?x a ?type. Filter not exists {?x a <http://www.coinsweb.nl/cbim-2.0.rdf#Concept>}. filter not exists{ ?x a owl:Ontology}
}
}
```

#name :Geen illegale RDF/RDFS/OWL objecten in een COINSContainer
 #description :only instances in a semantic BIM model
 #reference :COINS-NSWO
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
 PREFIX owl: <http://www.w3.org/2002/07/owl#>
 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

select distinct ?a ?b ?c

```
WHERE {Graph <instanceGraph>{
  ?a ?b ?c. filter (?c!=owl:Thing). filter (?c!=owl:Ontology). filter (?c!=owl:NamedIndividual). filter (?c!=owl:Nothing) . filter
  (STRSTARTS(STR(?c),"http://www.w3.org/1999/02/22-rdf-syntax-ns#")||((STRSTARTS(STR(?c),"http://www.w3.org/2000/01/rdf-
  schema#"))||(STRSTARTS(STR(?c),"http://www.w3.org/2002/07/owl#"))))
}
}
```

#name :geen Rdfs SubclassOf predicaten in een COINSContainer
 #description :No rdfs:subclassOf in a semantic BIM model
 #reference :COINS-NSCO
 #profile :COINS 2.0 Lite

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?a ?c
```

```
WHERE {Graph <instanceGraph>{
  ?a rdfs:subClassOf ?c
}
}
```

```
#name          :Geen rdfs:subPropertyOf predcaten een COINS container
#description    :No subPropertyOf in a semantic BIM model
#reference      :COINS-NSPO
#profile       :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?a ?c
```

```
WHERE {Graph <instanceGraph>{
  ?a rdfs:subPropertyOf ?c
}
}
```

```
#name          :valideren van disjoint individuals via Disjoint verklaringen op basis van owl:allDisjoint
#description    :Disjoint via owl:allDisjointClasses
#reference      :Cax-adx (listmod)
#profile       :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?x ?c1 ?c2
```

```
WHERE {Graph <schemaGraph>{
  ?y a owl:AllDisjointClasses .  ?y owl:members ?members .  ?members rdf:rest*/rdf:first ?c1 .  ?members
  rdf:rest*/rdf:first ?c2 .  FILTER (?c1 != ?c2) .
}
Graph <instanceGraph>{
  ?x a ?c1 .  ?x a ?c2
}
}
```

```
#name          :valideren van disjoint individuals via Disjoint verklaringen op basis van owl:disjointWith
#description    :Disjoint via owl:disjointWith
#reference      :cax-dw
#profile       :COINS 2.0 Lite
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
select distinct ?x ?c1 ?c2
```

```
WHERE {Graph <schemaGraph>{
  ?c1 owl:disjointWith ?c2
}
Graph <instanceGraph>{
  ?x a ?c1 .  ?x a ?c2 .
}
}
```