SEMANTIC DATA 2021
Solutions of suggested exercises

Practice 1. First order logic

1. Syntax of FOL

Suggested exercise 1 : which sentences are well formed FOL formulas or terms ?

Non logical symbols: constants a, b, functions f^1, g^2, predicates P^1, R^2, Q^3 (with indicated arity).

1) Q(a) — not well formed
2) P(y) — well formed
3) P(g(b)) — not well formed
4) ¬R(x, a) — well formed
5) Q(x, P(a), b) — not well formed
6) P(g(f(a), g(x, f(x)))) — well formed
7) Q(f(a), f(f(x)), f(g(f(z), g(a, b)))) — well formed
8) R(a, R(a, a)) — not well formed

Suggested exercise 2: find the free variables in the following formulas ?

1) P(x) ∧ ¬R(y, a) — x, y free
2) ∃x R(x, y) — y free
3) ∀x P(x) → ∃y ¬Q(f(x), y, f(y)) — x free in Q(f(x), y, f(y))
4) ∀x ∃y R(x, f(y)) — no free variable
5) ∀x ∃y R(x, f(y)) → R(x, y) — x, y free in R(x, y)

2. Finding the meaning of FOL formulas

Suggested exercise 1 : what is the meaning of the following formulas ?

1) ∀x [(StrongEngine(x) ∧ Car(x) ∧ Wheels(x, 4)) → Fast(x)]

   All four wheels cars with a strong engine are fast.

2) ∀x ∀y [(Parent(x, y) ∧ Ancestor(y)) → Ancestor(x)]

   Anybody who is the parent of an ancestor is also an ancestor.

3) ∀x ∀y [(Car(x) ∧ OnRoad(x, y) ∧ Highway(y) ∧ NormalConditions(y)) → FastSpeedAllowed(x)]

   For any car on any highway road under normal conditions, fast speed is allowed.

4) ∃t ∀p (¬Travel(t, p) ∨ FarFrom(p, Mycity))
   where travel(t, p) represents my travel to p at time t.

   Sometimes, either I don't travel anywhere or I travel far from the city I live in.

5) ∃t ∀p (Travel(t, p) → FarFrom(p, Mycity))

   Sometimes I travel far from the city I live in, if anywhere.

6) Are sentences 4 and 5 equivalent ?

   Yes (by definition of the implication).

3. Formulating sentences in FOL

Suggested exercise 1

The function *mapColor* and predicates *In(x, y)*, *Borders(x, y),* and *Country(x)* are given.
For each of the following sentences and corresponding candidate FOL expressions, indicate if the FOL expression
a) correctly expresses the English sentence;
b) is syntactically invalid and therefore meaningless; or
c) is syntactically valid but incorrect : does not express the meaning of the English sentence.

1) No region in South America borders any region in Europe.

   i. ¬[∃c ∃d (In(c, SouthAmerica) ∧ In(d, Europe) ∧ Borders(c, d))] correct
   ii. ∀c ∀d [In(c, SouthAmerica) ∧ In(d, Europe)] → ¬Borders(c, d)] correct
   iii. ¬∀c (In(c, SouthAmerica) → ∃d (In(d, Europe)∧ ¬Borders(c, d))) incorrect
   iv. ∀c (In(c, SouthAmerica) → ∀d (In(d, Europe) → ¬Borders(c, d))) correct

2) No two adjacent countries have the same map color. (this sentence requires equality).

   i. ∀x ∀ y (¬Country(x) ∨ ¬ Country(y)∨ ¬ Borders(x, y) ∨ ¬(mapColor(x) = mapColor(y))) correct
   ii. ∀x ∀ y ((Country(x) ∧ Country(y) ∧ Borders(x, y) ∧ ¬(x = y)) → ¬ (mapColor(x) = mapColor(y))) correct
   iii. ∀x ∀y (Country(x) ∧ Country(y) ∧ Borders(x, y) ∧ ¬(mapColor(x) = mapColor(y))) incorrect

Suggested exercise 2 : translate into FOL

1) Everyone is mad. ∀x mad(x)
2) There is at least one doctor. ∃x doctor(x)
3) Doctors are not lawyers. ∀x (doctor(x) → ¬lawyer(x))
4) Lawyers sue everyone. ∀x ∀y (lawyer(x) → sue(x, y))
5) Doctors sue back if they are sued. ∀x (doctor(x) → ∀y (sue(y, x)) → sue(x, y)))
6) There is an individual who does not sue.
   ∃x ¬∃y sue(x, y)
   [equivalent form: ∃x ∀y ¬sue(x, y)]

Suggested exercise 3 : define an appropriate language and translate the sentences in FOL :

1) Bill has at least one sister. $\exists x \, SisterOf(x, Bill)$
2) Bill has no sister. $\neg \exists x \, SisterOf(x, Bill)$
3) Every student takes at least one course.
   $\forall x \, (Student(x) \rightarrow \exists y \, (Course(y) \wedge Takes(x, y)))$
4) No student failed Geometry but at least one student failed Analysis.
   $\neg \exists x \, (Student(x) \wedge Failed(x, Geometry)) \wedge \exists x \, (Student(x) \wedge Failed(x, Analysis))$
5) Every student who takes Analysis also takes Geometry.
   $\forall x \, (Student(x) \wedge Takes(x, Analysis) \rightarrow Takes(x, Geometry))$

Suggested exercise 4 : in a world of labeled colored blocks, translate the following sentences in FOL :

1) A is above C, D is on E and above F.
   $Above(A, C) \wedge On(D, E) \wedge Above(E, F)$
2) A is green while C is not. $Green(A) \wedge \neg Green(C)$
3) Everything is on something. $\forall x \, \exists y \, On(x, y)$
4) Everything that is free has nothing on it. $\forall x \, (Free(x) \rightarrow \neg \exists y \, On(y, x))$
5) Everything that is green is free. $\forall x \, (Green(x) \rightarrow Free(x))$
6) There is something that is red and is not free. $\exists x \, (Red(x) \wedge \neg Free(x))$
7) Everything that is not green and is above B, is red.
   $\forall x \, (\neg Green(x) \wedge Above(x, B) \rightarrow Red(x))$

4. Manipulating formulas

Suggested exercise 1 : convert in conjunctive normal form

1) $\neg(\neg p \lor q) \lor (r \to \neg s)$

$\neg(\neg p \lor q) \lor (\neg r \lor \neg s)$

$(\neg\neg p \land \neg q) \lor (\neg r \lor \neg s)$

$(p \land \neg q) \lor (\neg r \lor \neg s)$ NNF

$(p \lor \neg r \lor \neg s) \land (\neg q \lor \neg r \lor \neg s)$ CNF

2) $p \to (q \land r)$

$\neg p \lor (q \land r)$ NNF

$(\neg p \lor q) \land (\neg p \lor r)$ CNF

3) $p \to (q \to r)$

$\neg p \lor \neg q \lor r$ NNF and CNF (viewed as a conjunction with only one conjunct)

4) $(p \to q) \to r$

$(\neg p \lor q) \to r$

$\neg(\neg p \lor q) \lor r$

$(p \land \neg q) \lor r$ NNF

$(p \lor r) \land (\neg q \lor r)$ CNF

5) $(\neg p \to (p \to q))$

$\neg\neg p \lor (\neg p \lor q)$

$p \lor \neg p \lor q$

True NNF and CNF

6) $(p \to (q \to r)) \to (p \to (r \to q))$

$\neg(\neg p \lor \neg q \lor r) \lor (\neg p \lor \neg r \lor q)$

$(p \land q \land \neg r) \lor (\neg p \lor \neg r \lor q)$ NNF

$(p \lor \neg p \lor \neg r \lor q) \land (q \lor \neg p \lor \neg r \lor q) \land (\neg r \lor \neg p \lor \neg r \lor q)$

$(\neg r \lor q) \land (q \lor \neg p \lor \neg r) \land (\neg r \lor \neg p \lor q)$

$q \lor \neg p \lor \neg r$ CCF

Suggested exercise 2 : convert in prenex conjunctive normal form

$\exists z\ (\exists x\ Q(x, z) \lor \exists x\ P(x)) \rightarrow \neg(\neg\exists x\ P(x) \land \forall x\ \exists z\ Q(z, x))$

$\exists z\ (\exists x\ Q(x, z) \lor \exists x\ P(x)) \rightarrow \neg(\neg\exists x\ P(x) \land \forall x\ \exists z\ Q(z, x))$ eliminating « → »

$\neg\ \exists z\ (\exists x\ Q(x, z) \lor \exists x\ P(x)) \lor \neg(\neg\exists x\ P(x) \land \forall x\ \exists z\ Q(z, x))$ importing negations

$\forall z\ \neg\ (\exists x\ Q(x, z) \lor \exists x\ P(x)) \lor (\neg\neg\exists x\ P(x) \lor \neg\forall x\ \exists z\ Q(z, x))$

$\forall z\ (\neg\exists x\ Q(x, z) \land \neg\exists x\ P(x)) \lor (\exists x\ P(x) \lor \exists x\ \forall z\ \neg\ Q(z, x))$

$\forall z\ (\forall x\ \neg Q(x, z) \land \forall x\ \neg P(x)) \lor (\exists x\ P(x) \lor \exists x\ \forall z\ \neg Q(z, x))$

$\forall z\ \forall x\ (\neg Q(x, z) \land \neg P(x)) \lor \exists x\ (P(x) \lor \forall z\ \neg Q(z, x))$ renaming to pull quantifiers in front

$\forall z\ \forall x\ \exists y\ \forall w\ ((\neg Q(x, z) \land \neg P(x)) \lor (P(y) \lor \neg Q(w, y)))$ prenex DNF

$\forall z\ \forall x\ \exists y\ \forall w\ ((\neg Q(x, z) \lor P(y) \lor \neg Q(w, y)) \land (\neg P(x) \lor P(y) \lor \neg Q(w, y)))$ prenex CNF

5. Interpretations and reasoning

Suggested exercise 1

given the model M defined by $\Delta = \{A, B, C\}$ and the interpretation function I :

$X^I = A$, $Y^I = A$, $Z^I = B$

$f^I = \{\langle A, B\rangle, \langle B, C\rangle, \langle C, C\rangle\}$

$P^I = \{A, B\}$

$Q^I = \{C\}$

$R^I = \{\langle B, A\rangle, \langle C, B\rangle, \langle C, C\rangle\}$

verify whether the following formulas are True :

1) $Q(f(Z))$ True
2) $R(X, Y)$ False
3) $\forall w\ R(f(w), w)$ True
4) $\forall u\ \forall v\ (R(u, v) \rightarrow \forall w\ R(u, w))$ False

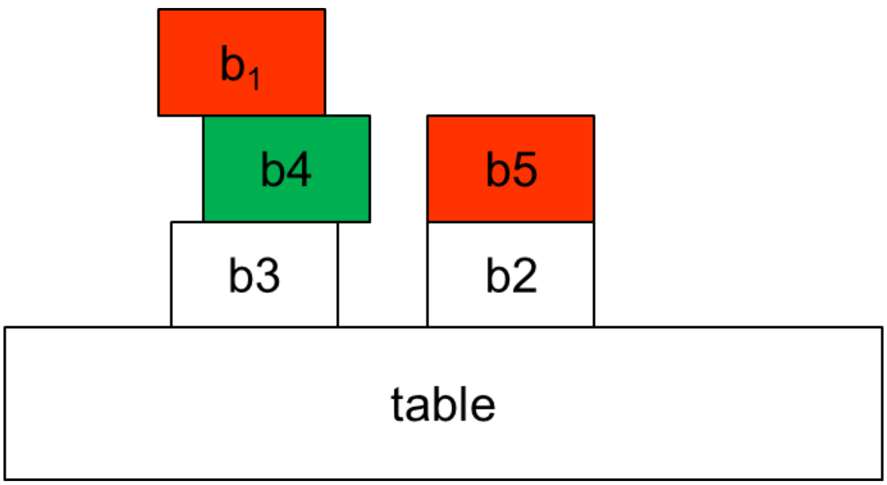
Suggested exercise 2 :

For the world of labeled colored blocks of suggested exercise 4 of section 3,

Consider the interpretation I defined by $A^I = b_1$, $B^I = b_2$, $C^I = b_3$, $D^I = b_4$, $E^I = b_5$
and by the picture below, where *On*, *Above, Green, Red* and *Free* have their normal meaning;

a) Complete the formal definition of that interpretation;

b) For each formula in suggested exercise 4 of section 3, determine whether it is satisfied of not by that interpretation.



a)

$On^{I} = \{<b_1, b_4>; <b_4, b_3>; <b_3, table>; <b_5, b_2>; <b_2, table>\}$

$Above^{I} = \{<b_1, b_4>; <b_1, b_3>; <b_1, table>; <b_4, b_3>; <b_4, table>; <b_3, table>; <b_5, b_2>; <b_5, table>; <b_2, table>\}$

$Free^{I} = \{<b_1>; <b_5>\}$, $Green^{I} = \{<b_4>\}$, $Red^{I} = \{<b_1>; <b_5>\}$

b) Formulas 1, 2, 3, 5 6 are False, 4 and 7 are True

Practice 2. RDF and RDFS

1. Manipulating the RDF Turtle language

Suggested exercise 1 : correct the mistakes in the following Turtle document :

```
PREFIX vcard: http://www.w3.org/2006/vcard/ns#
PREFIX foaf http://xmlns.com/foaf/0.1/

http://example.org/people/jlb#me a foaf:Person vcard:Individual

name "Jean-Louis Binot" . # Hint: foaf has this property

vcard:hasEmail  "jean-louis.binot@uliege.be";

vcard:hasTelephone  [ a vcard:Voice ,  vcard:hasValue  11111111] ;
```

Here is a corrected version :

```
PREFIX vcard: <http://www.w3.org/2006/vcard/ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

<http://example.org/people/jlb#me> a foaf:Person , vcard:Individual ;

        foaf:name "Jean-Louis Binot" ; # Hint: foaf has this property

        vcard:hasEmail  "jean-louis.binot@uliege.be";

        vcard:hasTelephone  [ a vcard:Voice ;  vcard:hasValue  11111111] .
```

Suggested exercise 2 : write an RDF model representing the following statements :

"Ideas and opinions" has been created by Einstein.
Book1 and Book2 have been created by the same unknown author.
Amazon states that "Ideas and opinions" has been published by Broadway Books.

Use the Dublin Core ontology and assume that an appropriate default namespace exist for the resource mentioned (e.g. *:Einstein)*.

The third statement requires to use reification. A possible solution is :

```
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
PREFIX : <http://example.org/myVocabulary/>

:Einstein  dc:Creator "Ideas and Opinions" .
:Book1 dc:Creator _:x.
:Book2 dc:Creator _:x.
:Amazon :states _:y.
_:y rdf:subject :BoradwayBooks.
_:y rdf:predicate dc:Publisher.
_:y rdf:object "Ideas and Opinions".
_:y rdf:type rdf:statement.
```

Suggested exercise 3 : express the following statements in RDF Turtle :

a) :

Georges lives in Brussels and Marie in a city located in Wallonia.

A possible solution is :

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

:Georges :livesin :Brussels .
:Marie :livesin :_x  .
 :_x a :city; :locatedin :Wallonia .
```

b)

Three students, Pierre, Jean and Marie are living in Liège, located in Belgium. They collectively follow the course of Logic, given by Willard Quine.

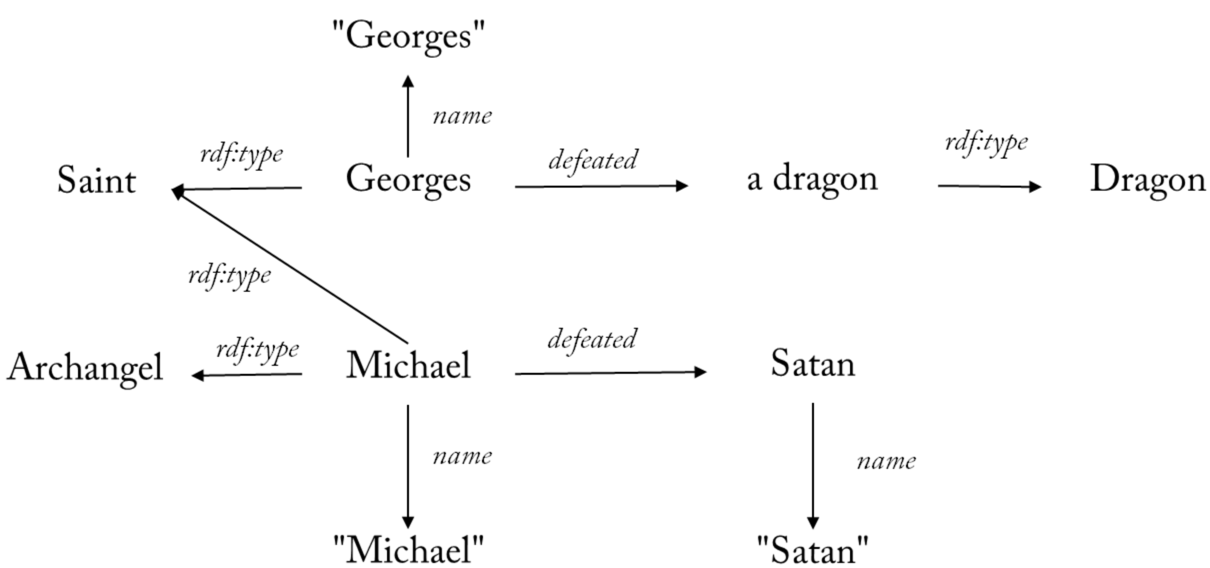
Each student lives individually in Liège but they follow collectively the course. A possible solution is to use a bag for the fact that they follow collectively the course :

```
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/myVocabulary/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>

:Pierre :livesin :Liège .
:Jean :livesin :Liège .
:Marie :livesin :Liège .
:Liège a :city; :locatedin :Belgium .
[ a rdf:Bag ;
rdf:_1 :Pierre ;
```

rdf:_2 :Jean ;
rdf:_3 :Marie
] :follows _:x .
_:x a :course ; dc:title "Logic" .
:WilliamQuine :gives _:x .

Suggested exercise 4 : model the following graph in RDF Turtle, using appropriate vocabularies and prefixes :



A possible solution is :

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://example.org/myVocabulary/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

:Georges a :Saint ; foaf:name "Georges" .
:Michael a :Archangel ; a :Saint ; foaf:name "Michael" .
:Satan foaf:name "Satan" .
:Michael :defeated :Satan .
:Georges :defeated :_x .
:_x a :Dragon .

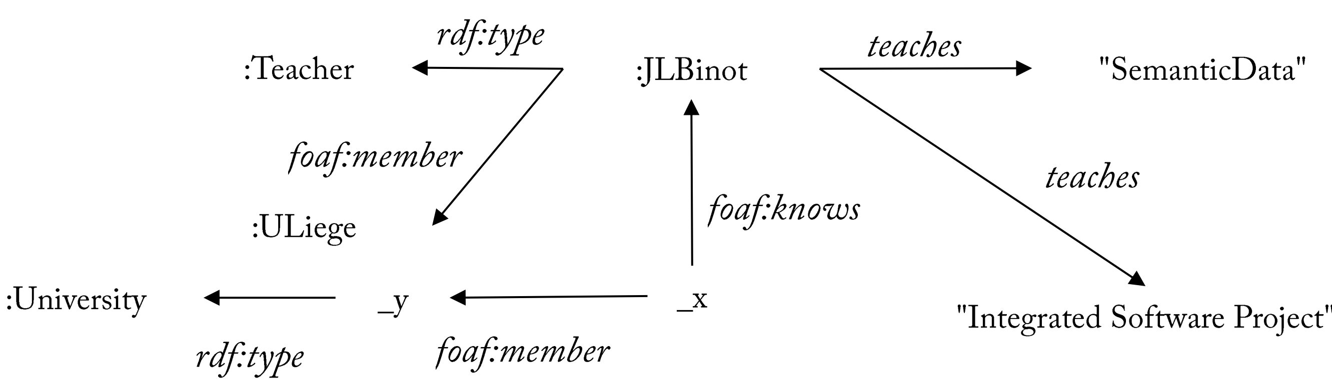
2. Transform triples in a labeled graph

Suggested exercise 1 : draw the graph corresponding to the following Turtle code :

```
PREFIX :    <http://example.org/>
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>

:JLBinot a :Teacher;
:teaches "SemanticData" , "Integrated Software Project" ;
foaf:member :ULiege .
[] foaf:knows :JLBinot; foaf:member [a :University] .
```

Solution :



3. Express a model in RDFS

Suggested exercise 1 : model a simple RDFS ontology about pizzas.

This exercise covers a part of the pizza ontology developed by University of Manchester. Students may refer to the full ontology, for which a pointer is posted on the course web page. A pointer towards a simpler RDFS version of that ontology corresponding to this exercise is also posted on the course web page.

Suggested exercise 1 : write an RDFS model about a family.

There are many ways to solve this. A solution is posted on the course web page. This solution assumes that classes such as Woman and Mother are not disjoint (somebody can be both a Woman and a Mother).

4. Applying RDFS inference rules

Suggested exercise 1 : which additional triplets can be inferred from the following RDFS code ?

```
PREFIX rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:  <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ex:   <http://example.org/>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/#>

ex:Person  a  rdfs:Class .
ex:Student  a rdfs:Class ;
      rdfs:subClassOf  ex:Person .
ex:Teacher  a rdfs:Class ;
      rdfs:subClassOf  ex:Person .
ex:John  ex:teaches ex:George .
ex:teaches  a           rdf:Property ;
      rdfs:domain     ex:Teacher;
      rdfs:range       ex:Student ;
      rdfs:subPropertyOf  foaf:knows .
```

Solution : the following triples can be inferred :

```
ex:George a ex:Student .
ex:George a ex:Person .
ex:John a ex:Teacher .
ex:John a ex:Person .
ex:John foaf:knows ex:George .
```

Practice 3. Description Logics

1. Translate DL into Natural Language and into FOL

Suggested exercise 1 look at the exercises 1, 2 and 3 i, j, k of the section "*Expressing concepts in description logics*" and try to express the same concepts in FOL.

From exercise 1 :

Person ⊓ Happy — Person(x) ∧ Happy(x)

Person ⊓ Happy ⊓ ∃owns.Pet — Person(x) ∧ Happy(x) ∧ ∃y ((owns(x, y) ∧ Pet(y))

Person ⊓ ∀owns.Cat — Person(x) ∧ ∀y(owns(x, y) → Cat(y))

Person ⊓ ¬Happy ⊓ ∃owns.(Cat ⊓ Old)

Person(x) ∧ ¬ Happy(x) ∧ ∃y ((owns(x, y) ∧ Cat(y) ∧ Old(y))

Person ⊓ ∃owns.Pet ⊓ ∀owns.(Cat ⊔ Fish)

Person(x) ∧ ∃y ((owns(x, y) ∧ Pet(y)) ∧ ∀z (owns(x, z) → Cat(z) ∨ Fish(z))

From exercise 2 :

Tree ⊆ Plant. — ∀x (Tree (x) → Plan(x))

Branch ⊆ ∀partOf.Tree — ∀x (Branch(x) → ∀y (partOf(x, y) → Tree(y))

Leaf ⊆ ∀partOf.Branch — ∀x (Leaf(x) → ∀y (partOf(x, y) → Branch(y))

Herbivore ≡ Animal ⊓ ∀eats.(Plant ⊔ ∀partOf.Plant)

∀x (Herbivore(x) ↔ Animal(x) ∧ ∀y (eats(x, y) → Plant(y) ∨ ∀z (partOf(y, z) → Plant(z))))

Carnivore ≡ Animal ⊓ ∃eats.Animal

∀x (Carnivore(x) ↔ Animal(x) ∧ ∃y (eats(x, y) ∧ Animal (y)))

Giraffe ⊆ Herbivore ⊓ ∀eats.Leaf

∀x (Girafe(x) → Herbivore(x) ∧ ∀y (eats(x, y) → Leaf(y)))

Lion ⊆ Carnivore ⊓ ∀eats.Herbivore

∀x (Lion(x) → Carnivore(x) ∧ ∀y (eats(x, y) → Herbivore(y)))

TastyPlant ⊆ Plant ⊓ ∃isEatenBy.Herbivore ⊓ ∃isEatenBy.Carnivore

∀x (TastyPlant(x) → Plant(x) ∧ ∃y (isEatenBy(x, y) ∧ Herbivore (y)) ∧ ∃z (isEatenBy(x, z) ∧ Carnivore(z)))

From exercise 3 :

(Bob, QE2) : controls — controls(Bob, QE2)

Bob : $(\exists controls.Car)$ — $\exists x\,(Car(x) \land controls(Bob, x))$

QE2 : $(Vehicle \sqcap \exists travelsOn.Water)$ — $Vehicle(QE2) \land \exists x\,(Water(x) \land travelsOn(QE2, x))$

2. Expressing concepts in description logics

Suggested exercise 1 : in description logics ALCN with concepts *Male*, *Female*, and a role *hasChild* define the following concepts (there are only people in *Male* and *Female*) :

a) Person — $Male \sqcup Female$

b) Mother — $Female \sqcap \exists hasChild.Person$

c) Parent — $Person \sqcap \exists hasChild.Person$ or $\exists hasChild.Person$

d) Childless — $Person \sqcap \neg\ \exists hasChild.Person$

e) Grandfather — $Male \sqcap \exists hasChild.Parent$

f) ParentOfSons (a parent with at least one son)
$Parent \sqcap \exists hasChild.Male$

g) ParentOfOnlySons — $Parent \sqcap \forall hasChild.Male$

h) MotherWithManyChildren (more than three children)
$Mother \sqcap \geq 3\ hasChild$

i) GrandfatherOfOnlyGrandsons
$Grandfather \sqcap \forall hasChild.(\ ParentOfOnlySons \sqcup Childless)$

Suggested exercise 2 :

a. In description logic ALC using concepts *Male*, *Doctor*, *Rich*, *Famous* and roles *hasChild*, *hasFriend*, define a concept HappyFather being a father whose all children are doctors and have rich or famous friends.

HappyFather ≡ Male ⊓ (∃hasChild.⊤) ⊓ ∀hasChild.(Doctor ⊓ ∃hasFriend.(Rich ⊔ Famous))

b. In description logic ALC using concepts *Female*, *Diplomat*, *StudyingAtUniversity*, *Working* and roles *married*, *hasChild*, define a concept SuccessfulMan being a man who married a diplomat and who has a child at university or having a job.

SuccessfullMan ≡ ¬ Female ⊓ ∃married.Diplomat ⊓ ∃hasChild.(StudyingAtUniversity ⊔ Working)

3. Interpretations / models

Suggested exercise 1 : is the following interpretation a model of the knowledge base from section 2 – exercise 4, repeated below ? Justify.

Please refer to the practice slides for the definition of the knowledge base and the interpretation.

Answer : yes.

In this model, there are more cars, more drivers, and more humans than what is strictly required by the knowledge base. Also, there are two cars sharing a wheel, and cars with a single wheel only, and there are no bicycles, leaving that axiom satisfied by default.

However, this interpretation clearly satisfies all statements in the knowledge base and is therefore a model of it.

Suggested exercise 2 : consider the following interpretation :

I : Δ^I = {t1, t2, f1, f2, p1, p2, j, k, l, m, n}

$Animal^I$ = {j, k, l, m, n}

$Plant^I$ = {t1, t2, f1, f2, p1, p2}

$Fern^I$ = {f1, f2}

$Tulip^I$ = {t1, t2}

$eats^I$ = {(j, f1), (k, f1), (k, t2), (l, p1), (l, p2), (m, p1), (m, t2), (n, f2), (n, p2)}

Find the interpretation in I of the following concepts:

a) ∃eats.Fern ⊓ ∃eats.Tulip

b) ∃eats.Fern ⊓ ∀eats.Fern

c) ∃eats.Fern ⊓ ∃eats.¬Fern

d) ∃eats.Plant ⊓ ∀eats.¬(Tulip ⊔ Fern)

Solution (note that if you are only given the above concepts you cannot assume that ferns and tulips are plants):

a) {k}

b) {j}

c) {k, n}

d) {l}

Suggested exercise 3 :

Suppose *Manager* and *Project* are concept names and *manages* is a role name. For each of the following expressions ϕ:

i. state whether it is an ALC concept, an ALC concept inclusion or none of the above.

ii. if ϕ is a concept inclusion, check whether ϕ follows from the empty TBox (i.e., ∅ ⊨ ϕ). If this is not the case, define an interpretation I such that I ⊭ ϕ.

if ϕ is a concept, check whether ϕ is satisfiable. If so, define an interpretation I such that $\phi^I \neq \emptyset$.

a) *Manager ⊆ ∃ manages.⊥*

i) concept inclusion.

ii) is not valid : Δ^I={a,b}; $manages^I$ ={(a,b)}; $Manager^I$ ={a}.

In fact, *∃ manages.⊥* is unsatisfiable (always empty!)

b) (≥7 manages. T) ⊆ Manager

states that anybody who manages at least 7things is a manager.

i) not an ALC concept nor ALC concept inclusion (because of the numerical quantification). It is however a concept inclusion in an extension of ALC.

ii) is not valid : $\Delta^I=\{a,b_1, \dots , b_7\}$; $Manager^I = \emptyset$; $manages^I =\{(a,b_1), \dots, (a, b_7\ \}$

c) (≥8 manages.Project) ⊆ Manager

states that anybody who manages at least 8 projects is a manager.

i) not an ALC concept nor ALC concept inclusion. It is however a concept inclusion in an extension of ALC.

ii) is not valid : $\Delta^I=\{a,b_1, \dots , b_8\}$; $Manager^I = \emptyset$; $manages^I =\{(a,b_1), \dots, (a, b_8\}$; $Project^I = \{ b_1, \dots , b_8\}$

d) *∀manages. T ⊆ ∃manages.Project*

states "everything manages a project" because *∀manages. T* = Δ^I for all interpretations.

i) concept inclusion.

ii) is not valid : Δ^I={a}; $manages^I = \emptyset$, $Project^I = \emptyset$

e) *∃manages. T ⊆ (≥ 4 manages. T)*

i) concept inclusion.

ii) is not valid : Δ^I={a, b}; $manages^I$ ={(a,b)} .

f) *(≥ 4 manages. T) ⊆ ∃manages. T*

states that "everybody managing at least four things manages something".

i) concept inclusion.

ii) it is valid.

Practice 7. SPARQL

2. Querying DBpedia

In the following exercises all namespace declarations are already defined in the SNORQL endpoint, except the one for dbo : which must be added.

DBPedia SNORQL endpoint to test DBPedia queries : http://dbpedia.org/snorql/
(select XML output).

Suggested exercise 1

a) Find all information on Steve Jobs.

```
describe :Steve_Jobs
```

b) Find 100 distinct names of persons in DBpedia.

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT  DISTINCT ?name ?x WHERE {
  ?person a dbo:Person  .
  ?person foaf:name ?name .
} LIMIT 100
```

c) Find in alphabetic order all names of Belgian soccer players in DBpedia. Add their position on the field.

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT * WHERE {
  ?player a dbo:SoccerPlayer .
  ?player dbo:birthPlace :Belgium .
  ?player dbo:position ?position .
  ?player foaf:name ?name .
} ORDER BY ASC(?name)
```

Suggested exercise 2

a) Find all books published in 1970.

```
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT * WHERE
{
?book a dbo:Book .
?book dbpedia2:pubDate 1970 .
?book dbo:publisher ?publisher .
}
```

b) Find all books authored by Jules Verne, and their publisher.

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT * WHERE
{
?book a dbo:Book .
?book dbo:author :Jules_Verne .
?book dbo:publisher ?publisher .
}
```

c) For the books found in b), find their publication date and sort and count them by publication date.

```
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT  ?date (COUNT(?book) AS ?count)
 WHERE
{
?book a dbo:Book .
?book dbo:author :Jules_Verne .
?book dbpedia2:pubDate ?date .
} ORDER BY ASC(?date)
```

Practice 8. Reasoning with description logics

1. Structural subsumption

Suggested exercise 1 : check if the following subsumptions are valid

a) Adult ⊓ (Male ⊓ Rich) ⊆ Adult ⊓ Male

Normalisation : Adult ⊓ Male ⊓ Rich ⊆ Adult ⊓ Male

Recursive comparison for each element on the right side :

- Adult is an atomic concept name and is present on the left side
- Male is an atomic concept name and is present on the left side

The subsumption is correct.

b) Adult ⊓ Male ⊓ ∀child.Rich ⊆ Adult ⊓ ∀child.Male ⊓ ∀child.Rich

Normalisation : Adult ⊓ Male ⊓ ∀child.Rich ⊆ Adult ⊓ ∀child.(Male ⊓ Rich)

Recursive comparison for each element on the right side :

- Adult is an atomic concept name and is present on the left side
- ∀child is a universal restriction present on both sides. We need to check if Male ⊓ Rich subsumes Rich : Rich ⊆ Male ⊓ Rich
  Male is an atomic concept name on the right side which is not present on the left side.

The subsumption is not correct.

2. Transformation in negative normal form

Suggested exercise 2 : transform the following GCIs into Negative Normal Form

a) Human ⊆ ∃hasParent.Human

The GCI needs to be transformed into a disjunction before calling the tableau algorithm :

¬Human ⊔ ∃hasParent.Human

That expression is in negative normal form.

b) Orphan ⊆ Human ⊓ ¬∃hasParent.Alive

The GCI needs to be transformed into a disjunction :

¬Orphan ⊔ Human ⊓ ¬∃hasParent.Alive

The negation then needs to be pushed inwards :

¬Orphan ⊔ Human ⊓ ∀hasParent.¬ Alive

That expression is in negative normal form.

c) Professor ⊆ ¬(¬Person ⊔ (¬UniversityMember ⊔ Student))

The GCI needs to be transformed into a disjunction :

¬Professor ⊔ ¬(¬Person ⊔ (¬UniversityMember ⊔ Student))

The negation then needs to be pushed inwards :

¬Professor ⊔ (Person ⊓ ¬ (¬UniversityMember ⊔ Student))

¬Professor ⊔ (Person ⊓ (UniversityMember ⊓ ¬Student))

That expression is in negative normal form.

3. Using the ALC tableau algorithm

<u>**Suggested exercise 1 :** check whether the following subsumption</u>

Vegan ⊑ Vegetarian

is entailed by the Tbox

TBox =

{ Vegan ≡ Person ⊓ ∀eats.Plant;

Vegetarian ≡ Person ⊓ ∀eats.(Plant ⊔ Dairy) }

We will need to show that Vegan ⊓ ¬Vegetarian is unsatisfiable.

We need first to eliminate the TBox by unfolding the concept :

Vegan ⊓ ¬Vegetarian ≡ *Person ⊓ ∀eats.Plant ⊓ ¬ (Person ⊓ ∀eats.(Plant ⊔ Dairy))*

Next step : convert the formula in negative normal form :

Person ⊓ ∀eats.Plant ⊓ (¬Person ⊔ ¬ ∀eats.(Plant ⊔ Dairy))

Person ⊓ ∀eats.Plant ⊓ (¬Person ⊔ ∃eats.(¬Plant ⊓ ¬Dairy))

Next step, call the algorithm with

A_0 = { *x : (Person ⊓ ∀eats.Plant ⊓ (¬Person ⊔ ∃eats.(¬Plant ⊓ ¬Dairy)))*}

Application of the rules :

$\rightarrow_\sqcap$ $A_1 = A_0$ ∪ {x : Person; x : (∀eats.Plant ⊓ (¬Person ⊔ ∃eats.(¬Plant ⊓ ¬Dairy)))}

$\rightarrow_\sqcap$ $A_2 = A_1$ ∪ {x : ∀eats.Plant; x : (¬Person ⊔ ∃eats.(¬Plant ⊓ ¬Dairy))}

$\rightarrow_\sqcup$ b_1) $A_3 = A_2$ ∪ { x : ¬Person } clash

b_2) $A_3 = A_2$ ∪ { x : ∃eats.(¬Plant ⊓ ¬Dairy)}

$\rightarrow_\exists$ $A_4 = A_3$ ∪ { (x, y) : eats; y : (¬Plant ⊓ ¬Dairy) }

$\rightarrow_\forall$ $A_5 = A_4$ ∪ { y : Plant } (because (x, y) : eats)

$\rightarrow_\sqcap$ $A_6 = A_5$ ∪ {y : ¬Plant ; y : ¬Dairy } clash

All branches have led to clashes and there are no rules left to apply. The input concept is unsatisfiable; hence the initial subsumption holds.

Suggested exercise 2

check whether the ABox

A = {a : A ⊓ ∃s.F; (a, b) : s; a : ∀s.(¬F ⊔ ¬B); (a, c) : r; b : B; c : C ⊓ ∃s.D}

is satisfiable. If so, indicate a satisfying model for it.

As there is already an ABox with assertions, there is no need to add an assertion to it.

The concepts in the ABox are already in negative normal form.

We can apply the rules in the following steps :

A_0 = { a : A ⊓ ∃s.F; (a, b) : s; a : ∀s.(¬F ⊔ ¬B); (a, c) : r; b : B; c : C ⊓ ∃s.D }

$\rightarrow_\sqcap$ $A_1 = A_0$ ∪ {a : A; a: ∃s.F }

$\rightarrow_\exists$ $A_2 = A_1$ ∪ { (a, x) : s; x : F }

$\rightarrow_\forall$ $A_3 = A_2$ ∪ { b : ¬F ⊔ ¬B } (because (a, b) : s)

$\rightarrow_{\forall}$ $A_4 = A_3 \cup \{ x : \neg F \sqcup \neg B \}$ (because (a, x) : s)

$\rightarrow_{\sqcup}$ a) $A_5 = A_4 \cup \{ x : \neg F \}$ clash

b) $A_5 = A_4 \cup \{ x : \neg B \}$

$\rightarrow_{\sqcup}$ b_1) $A_6 = A_5 \cup \{ b : \neg F \}$

$\rightarrow_{\sqcap}$ $A_7 = A_6 \cup \{c : C; c: \exists s.D \}$

$\rightarrow_{\exists}$ $A_8 = A_7 \cup \{ (c, y) : s; y : D \}$ no more rule to apply and no clash : A_0 is consistent.

b_2) $A_6 = A_5 \cup \{ b : \neg B \}$ no need to try that branch as we have found a consistent solution. (if we check it, it will yield a clash).

A satisfying model is : $\Delta^I = \{a, b, c, x, y\}$, $A^I = \{a\}$, $B^I = \{b\}$, $C^I = \{c\}$, $F^I = \{x\}$, $D^I = \{y\}$, $r^I = \{(a, c)\}$, $s^I = \{(a, b), (a, x), (c, y)\}$

Suggested exercise 3

Given the knowledge base K with

$T = \{A \sqsubseteq B, B \sqsubseteq C, C \sqsubseteq \exists r.D, D \sqsubseteq \neg A, A \sqsubseteq \forall r.A \}$

$A = \{a : A, c : D, (a, b) : r, (b, c) : r\}$

Check if K is consistent and if so provide a model for K.

$A_0 = \{a : A; c : D; (a, b) : r; (b, c) : r\}$

$\rightarrow_{GCI}$ $A_1 = A_0 \cup \{a : B\}$

$\rightarrow_{GCI}$ $A_2 = A_1 \cup \{a : C\}$

$\rightarrow_{GCI}$ $A_3 = A_2 \cup \{a : \exists r.D\}$

$\rightarrow_{\exists}$ $A_4 = A_3 \cup \{(a, x) : r; x : D\}$

$\rightarrow_{GCI}$ $A_5 = A_4 \cup \{x : \neg A\}$

$\rightarrow_{GCI}$ $A_6 = A_5 \cup \{c : \neg A\}$

$\rightarrow_{GCI}$ $A_7 = A_6 \cup \{a : \forall r.A \}$

$\rightarrow_{\forall}$ $A_8 = A_7 \cup \{b : A\}$

$\rightarrow_{GCI}$ $A_9 = A_8 \cup \{b : B; b : C\}$

$\rightarrow_{\exists}$ $A_{10} = A_9 \cup \{(b, y) : r; y : D\}$ y blocked by x

$\rightarrow_{\forall}$ $A_{11} = A_7 \cup \{c : A\}$ clash

$\rightarrow_{\forall}$ $A_{11} = A_7 \cup \{x : A\}$ clash

No other branch; K is inconsistent.