Knowledge Representation and Semantic Technologies – 14/6/2017

LAST NAME: ____________________ FIRST NAME: ____________________ ID (MATRICOLA): __________	Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.dis.uniroma1.it/~rosati/gd`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, ____________________

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$\begin{array}{rcl} A & \sqsubseteq & B \sqcup C \\ B & \sqsubseteq & \exists R.D \\ C & \sqsubseteq & \forall R.\neg D \\ A & \sqsubseteq & \forall R.E \\ D \sqcap E & \sqsubseteq & \neg B \end{array}$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept A is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where the interpretation of A is non-empty;

(c) tell whether the concept $B \sqcap C$ is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where the interpretation of B is non-empty;

(d) given the ABox $\mathcal{A} = \{A(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $\exists R.E(a)$, explaining your answer.

Exercise 2 Given the following ASP program P:

```
r(x,y) :- p1(y), p2(x).
s(x,y) :- q(x,y).
s(x,y) :- q(x,z), s(z,y).
t(x,y) :- r(x,y), not s(x,y).
t(x,y) :- s(x,y), not r(x,y).
v(x,y) :- t(x,y).
v(x,y) :- t(y,x).
w(x,y) :- v(x,y), not r(x,y).
p1(c). p1(d). p2(a). p2(b). p2(c).
q(a,b). q(b,c). q(c,d). q(d,e). q(e,d).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

We want to formalize knowledge about the domain of students and professors. In particular, we want to formalize the following statements:

1. every student is a person;
2. every professor is a person;
3. busy person is a subclass of person;
4. the property "is friend of" has domain person and range person;
5. the property "studies with" has domain student and range student;
6. the property "studies with" is a subproperty of the property "is friend of";
7. every professor is a friend of at least one professor;
8. every student studies with at least one student;
9. everyone who is both a student and a professor is a busy person.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following: $\mathcal{ALC}$, Datalog, ASP, OWL, *DL-Lite$_R$*, $\mathcal{EL}$, *RL*, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

(a) Write an RDF/RDFS model representing the following statements about URIs `Person`, `Director`, `Actor`, `Writer`, `Movie`, `Country`, `Comedy`, `Drama`, `Man`, `Woman`, `filmedIn`, `hasBoxOfficeGross`, `isDirectorOf`, `isWriterOf`, `actsIn`, `bornIn`, `Joe`, `Mary`, `Ann`, `Paul`, `Italy`, `France`, `ABC`, `XYZ`.

1. `Person`, `Director`, `Writer`, `Actor`, `Country`, `Movie`, `Comedy`, `Drama`, `Man`, and `Woman` are classes;

2. `Man` and `Woman` are subclasses of `Person`;
3. `Comedy` and `Drama` are subclasses of `Movie`;
4. `actsIn`, `bornIn`, `filmedIn`, `isDirectorOf` and `isWriterOf` are properties;
5. `isDirectorOf` has domain `Director` and range `Movie`;
6. `flmedIn` has domain `Movie` and range `Country`;
7. `bornIn` has domain `Person` and range `Country`;
8. `actsIn` has domain `Actor` and range `Movie`;
9. `hasBoxOfficeGross` has domain `Movie` and range `xsd:integer`;
10. Ann is the director and the writer of movie XYZ;
11. Joe and Paul act in movie ABC;
12. ABC was filmed in France;
13. Ann is a woman;
14. Paul is a man.

(b) Write SPARQL queries corresponding to the following requests: (b1) return all the pairs of movies having the same writer and such that at least one actor acts in both movies; (b2) return the directors of comedies filmed in Italy, and, optionally, the country where the director was born.

Exercise 5

(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

(b) Add to the above ontology the axioms formalizing the following statements:

1. add a new property `isWrittenBy` and state that it is the inverse of `isWriterOf`;
2. add a new class `WrittenByMultipleAuthors` and state that it corresponds to the class of movies written by at least two writers;
3. add a new class `LowBudgetMovie` and state that it corresponds to the class of movies played by at most 5 actors;
4. add the new class `ItalianMovie` and state that such a class corresponds to the class consisting of every movie such that all its writer(s) and director(s) were born in Italy.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning (that is, the models) of the ontology? if so, identify and list such axioms.

Exercise 6

(a) Axiomatize the following scenario, appropriately with action precondition and effect axioms, and obtain successor state axioms.

Fluents:

- `boxClosed(s)` - The box is closed in situation s.
- `robotCloseToBox(s)` - The robot is close to the box in situation s.
- `objectInsideBox(s)` - The object is inside the box in situation s.

Actions:

- `moveCloseToBox` - The robot moves close to the box. This can be done if the robot is not close to the box, and has the effect that the robot will be close to the box.
- `openBox` - The robot opens the box. This can be done if the robot is close to the box and the box is closed, and has the effect that the box will be open (that is, not closed).
- `extractObject` - The robot extracts the object inside the box. This requires that the box is open and the robot is close to the box, and has the effect that the object will be outside the box.

Initial situation description: Initially the robot is not close to the box, the box is closed, and the object is inside the box.

(b) Show, by applying regression, that the object will be outside the box after the sequence of actions `moveCloseToBox`, `openBox`, `extractObject`, and that the sequence of actions is indeed executable.

1(a) Is T satisfiable?

We look for an interpretation I s.t. $A^I \subseteq B^I \sqcup C^I$

$B^I \subseteq (\exists.R.D)^I$

$C^I \subseteq (\forall R.\neg D)^I$

$A^I \subseteq (\forall R.E)^I$

$(D \sqcap E)^I \subseteq (\neg B)^I$

Idea: let's define

$A^I = B^I = C^I = (D\sqcap E)^I = \emptyset$

Then I: $\begin{cases} \Delta^I = \{d\} \\ A^I = B^I = C^I = (D\sqcap E)^I = \emptyset \\ R = \emptyset \end{cases}$ ✓

I satisfies the containment conditions, therefore I is a model of T.

(b). Is A satisfiable wrt T? $A^I \neq \{\}$

We modify the previous interpretation I:

$A^I = \{d\} \rightarrow B \sqcup C = \{d\} \rightarrow B = \{d\}$

$B = \{d\} \rightarrow \exists R.D = \{d\}$

$\hookrightarrow \{d \in \Delta^I \mid \exists e \in \Delta^I \text{ s.t. } (d,e) \in r^I \text{ AND } e \in D^I\} \rightarrow r = \{(d,e)\}$

$D = \{e\}$

$C = \emptyset$

$\Delta = \{d, e\}$

$A = \{d\} \rightarrow \forall R.E = \{d, e\}$

$\hookrightarrow \{d \in \Delta^I \mid \forall e \in \Delta, \text{ IF } (d,e) \in r^I \text{ THEN } e \in E^I\}$

$\hookrightarrow E^I = \{e\}$

$(D \sqcap E)^I = \{e\}$

$I' \begin{cases} A^{I'} = \{d\} & (D\sqcap E)^I = \{e\} \\ B^{I'} = \{d\} & \Delta^{I'} = \{d, e\} \\ C^{I'} = \emptyset & r^{I'} = \{(d,e)\} \end{cases}$

c) Is $B \sqcap C$ satisfiable wrt T?

$B \sqcap C$ sat means that there is element both in C and in B

$B^{I''} = \{d\}$ ✓

$C^{I''} = \{d\}$ ✓ But $B^{I''} \subseteq (\exists.R.D)^{I''}$ ✓

$C^{I''} \subseteq (\forall R.\neg D)^{I''}$

Notice that $\exists.R.D$ and $\forall R.\neg D$ are opposite so both conditions will be never verified. Then $B \sqcap C$ is never satisfied

d)

$A^I \subseteq B^I \cup C^I$

$B^I \subseteq (\exists.R.D)^I$

$C^I \subseteq (\forall R.\neg D)^I$

$A^I \subseteq (\forall R.E)^I$

$(D \sqcap E)^I \subseteq (\neg B)^I$

$A = \{A(a)\}$

Does $K = \langle T, A \rangle$ entails $\exists R.E(a)$?

Tableau starts with:

$A_0 = A \cup \{\neg \exists R.E(a)\} = \{\underline{A(a)}, \forall R.\neg E(a)\}$

$C_{GC1} = (\neg A \sqcup B \sqcup C) \sqcap (\neg B \sqcup \exists R.D) \sqcap (\neg C \sqcup \forall R.\neg D) \sqcap (\neg A \sqcup \forall R.E) \sqcap (\neg D \sqcup \neg E \sqcup \neg B)$

C_{GC1} rule: $A_1 = A_0 \cup \{C_{GC1}(a)\}$

$\sqcap$ rule: $A_2 = A_1 \cup \{(\neg A \sqcup B \sqcup C)(a), (\neg B \sqcup \exists R.D)(a), (\neg C \sqcup \forall R.\neg D)(a), (\neg A \sqcup \forall R.E)(a), (\neg D \sqcup \neg E \sqcup \neg B)(a)\}$

Rule: $A_3 = A_2 \cup \{\neg A(a)\}$ | $A_4 = A_2 \cup \{B(a)\}$ | $A_5 = A_2 \cup \{C(a)\}$

CLASH

$A_{14}: A_5 \cup \{\neg A(a)\}$ | $A_{15}: A_5 \cup \{\forall z.E(a)\}$

CLASH

$A_6 : A_4 \cup \{\neg B(a)\}$ | $A_7 : A_4 \cup \{\exists z.D(a)\}$

CLASH

$A_{16}: A_{15} \cup \{\neg C(a)\}$ | $A_{17} : A_{15} \cup \{\forall z.\neg D(a)\}$

CLASH

$A_8 : A_7 \cup \{z(a,x_1), D(a)\}$

$A_{18} : A_{17} \cup \{\neg B(a)\}$ | $A_{19} : A_{17} \cup \{\exists z.D(a)\}$

$A_9 : A_8 \cup \{\neg E(a)\}$

$A_{10} : A_9 \cup \{\neg A(a)\}$ | $A_{11} : A_9 \cup \{\forall z E(a)\}$

CLASH

$A_{20} : A_{18} \cup \{\neg D(a)\}$ | $A_{21} : A_{18} \cup \{\neg E(a)\}$ | $A_{22} : A_{18} \cup \{\neg B(a)\}$

OPEN | OPEN | OPEN

$A_{12} : A_{11} \cup \{E(x_1)\}$

$A_{13} : A_{12} \cup \{\neg E(x_1)\}$

CLASH

$A_{23} : A_{19} \cup \{z(a,x_1), D(a)\}$

$A_{24} : A_{23} \cup \{E(x_1)\}$

$A_{25} : A_{24} \cup \{\neg E(x_1)\}$

CLASH

A_{20}, A_{21}, A_{22} don't clash

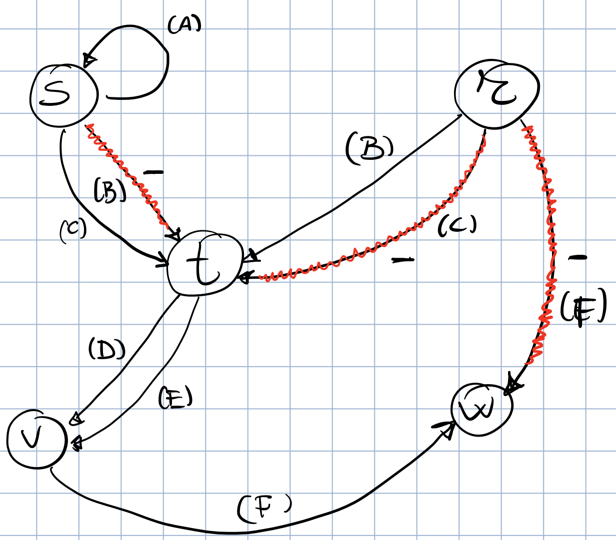
so $\exists R.E(a)$ is not entailed

2

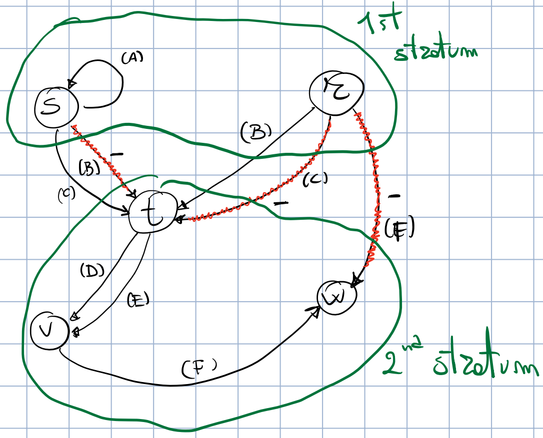
```
r(x,y) :- p1(y), p2(x).
s(x,y) :- q(x,y).
s(x,y) :- q(x,z), s(z,y). (A)
t(x,y) :- r(x,y), not s(x,y). (B)
t(x,y) :- s(x,y), not r(x,y). (C)
v(x,y) :- t(x,y). (D)
v(x,y) :- t(y,x). (E)
w(x,y) :- v(x,y), not r(x,y). (F)
p1(c). p1(d). p2(a). p2(b). p2(c).
q(a,b). q(b,c). q(c,d). q(d,e). q(e,d).
```

no cycle with negative edges

so T is stratified



```
r(x,y) :- p1(y), p2(x).
s(x,y) :- q(x,y).
s(x,y) :- q(x,z), s(z,y).  (A)
t̄(x,y) :- r(x,y), not s(x,y). (B)
t(x,y) :- s(x,y), not r(x,y).  (C)
v(x,y) :- t(x,y). (D)
v(x,y) :- t(y,x). (E)
w̄(x,y) :- v(x,y), not r(x,y). (F)
p1(c). p1(d). p2(a). p2(b). p2(c).
q(a,b). q(b,c). q(c,d). q(d,e). q(e,d).
```



$$MM_0 = \{p_1(c), p_1(d), p_2(a), p_2(b), p_2(c), q(a,b), q(b,c), q(c,d), q(d,e), q(e,d)\}$$

$$T_{P_1}(MM_0) = \{r(a,c), r(b,c), r(a,d), r(b,d), s(a,b), s(b,c), s(c,d), s(d,e), s(e,d)\} = I_1$$

$$T_{P_1}(I_1 \cup MM_0) = \{s(b,d), s(c,e), s(d,d), s(e,e), s(a,c)\} = I_2$$

$$T_{P_1}(I_2 \cup MM_0) = I_2 \quad \text{fixpoint} \rightarrow MM_1 = MM_0 \cup I_2$$

$$T_{P_2}(MM_1) = \{t(a,d), t(a,b), t(c,d), t(d,e), t(e,d), t(c,e), t(d,d), t(e,e)\} = I_3$$

$$T_{P_2}(I_3 \cup MM_1) = \{v(a,d), v(a,b), v(c,d), v(d,e), v(e,d), v(c,e), v(d,d), v(e,e)\} = I_4$$

$$T_{P_2}(I_4 \cup MM_1) = \{v(d,a), v(b,a), v(d,c), v(e,c)\} = I_5$$

$$T_{P_3}(I_5 \cup MM_1) = \{w(a,b), w(c,d), w(d,e), w(e,d), w(c,e), w(d,d), w(e,e), w(d,a), w(b,a), w(d,c), w(e,c)\} = I_6$$

I_6 is the answer set of P

(4)

```
@prefix rdf: <http://...>.
@prefix rdfs: <https://...>.
@prefix myns: <https://...>.
(1)
myns:Person rdf:type rdfs:Class.
myns:Director rdf:type rdfs:Class.
myns:Writer rdf:type rdfs:Class.
myns:Actor rdf:type rdfs:Class.
myns:Country rdf:type rdfs:Class.
myns:Movie rdf:type rdfs:Class.
myns:Comedy rdf:type rdfs:Class.
myns:Drama rdf:type rdfs:Class.
myns:Man rdf:type rdfs:Class.
myns:Woman rdf:type rdfs:Class.
(2)
myns:Man rdfs:subClassOf myns:Person.
myns:Woman rdfs:subClassOf myns:Person.
(3)
myns:Comedy rdfs:subClassOf myns:Movie.
myns:Drama rdfs:subClassOf myns:Movie.
(4)
myns:actsIn rdf:type rdf:property.
myns:bornIn rdf:type rdf:property.
myns:filmedIn rdf:type rdf:property.
myns:isDirectorOf rdf:type rdf:property.
myns:isWriterOf rdf:type rdf:property.
(5)
myns:isDirectorOf rdfs:domain myns:Director.
myns:isDirectorOf rdfs:range myns:Movie.
(6)
myns:filmedIn rdfs:domain myns:Movie.
myns:filmedIn rdfs:range myns:Country.
(7)
myns:bornIn rdfs:domain myns:Person.
myns:bornIn rdfs:range myns:Country.
(8)
myns:actsIn rdfs:domain myns:Actor.
myns:actsIn rdfs:range myns:Movie.
(9)
myns:hasBoxOfficeGross rdfs:domain myns:Movie.
myns:hasBoxOfficeGross rdfs:range xsd:integer.
(10)
myns:Ann myns:isDirectorOf myns:XYZ.
myns:Ann myns:isWriterOf myns:XYZ.
(11)
myns:Joe myns:actsIn myns:ABC.
myns:Paul myns:actsIn myns:ABC.
(12)
myns:ABC myns:filmedIn myns:France.
(13)
myns:Ann rdf:type myns:Woman.
(14)
myns:Paul rdf:type myns:Man.
```

In this case use DataPropertyRange

```
(b1)
PREFIX myns: <https://...>
SELECT ?x ?y
WHERE {?z myns:isWriterOf ?x.
      ?z myns:isWriterOf ?y.
      ?w myns:actsIn ?x.
      ?w myns:actsIn ?y.}
(b2)
PREFIX myns: <https://...>
SELECT ?x ?w
WHERE {?x myns:isDirectorOf ?y.
      ?y rdf:type  myns:Comedy.
      ?y myns:filmedIn myns:Italy.
      OPTIONAL{?x myns:bornIn ?w. }}
```

RDFS to OWL

(5a)

```
Declaration(Class(myns:Person))
SubClassOf(myns:Men myns:Person))
Declaration(ObjectProperty(myns:actsIn))
ObjectPropertyDomain(myns:isDirectorOf myns:Director))
ObjectPropertyRange(myns:isDirectorOf myns:Movie))
ObjectPropertyAssertion(myns:isDirectorOf myns:Ann myns:XYZ))
ClassAssertion(myns:Women myns:Ann)
```

(5b)

```
• Declaration(ObjectProperty(myns:isWrittenBy))
  InverseObjectProperties(myns:isWrittenBy myns:isWriterOf)
• Declaration(Class(myns:WrittenByMultipleAuthors))
  EquivalentClasses(myns:WrittenByMultipleAuthors
      ObjectIntersectionOf(myns:Movies
          ObjectSomeValuesFrom(
              ObjectMinCardinality(2
                  myns:isWrittenBy myns:Writer))))

• Declaration(Class(myns:LowBudgetMovie))
  EquivalentClasses(myns:LowBudgetMovie
      ObjectIntersectionOf(myns:Movies
          ObjectSomeValuesFrom(
              ObjectMaxCardinality(5
                  ObjectInverseOf(myns:actsIn)
                  myns:Actor))))
```

③

1. every student is a person;
2. every professor is a person;
3. busy person is a subclass of person;
4. the property "is friend of" has domain person and range person;
5. the property "studies with" has domain student and range student;
6. the property "studies with" is a subproperty of the property "is friend of";
7. every professor is a friend of at least one professor; range
8. every student studies with at least one student;
9. everyone who is both a student and a professor is a busy person.

$\exists isFriendOf \sqsubseteq Person$ DOMAIN.

$Person \sqsubseteq \exists isFriendOf^{-}$ RANGE

$stW \sqsubseteq isFriendOf$ (NOT IN ALC)

$PROFESSOR \sqsubseteq \exists isFriendOf.PROFESSOR$

$STUDENT \sqsubseteq \exists studiesWith.STUDENT$

$STUDENT \sqcap PROF \sqsubseteq BUSYP$

in OWL

SubClassOf(myns:Professor
ObjectSomeValuesFrom(myns:hasFriend
myns:Professor))

SubClassOf(
ObjectIntersectionOf(myns:Student
myns:Professor)
myns:BusyPerson)

$\mathcal{ALC}$, Datalog, ASP, OWL, $DL\text{-}Lite_R$, $\mathcal{EL}$, RL, RDFS

	$\mathcal{ALC}$	Datalog	ASP	OWL	$DL\text{-}Lite_R$	$\mathcal{EL}$	RL	RDFS
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	—	X
5	X	X	X	X	X	X	—	X
6	—	X	X	X	X	—	X	X
7	X	—	—	X	—	X	X	—
8	X	—	—	X	—	X	X	—
9	X	—	—	X	—	X	X	—
	NO	NO	NO	YES	NO	NO	NO	NO

Knowledge Representation and Semantic Technologies – 21/7/2017

LAST NAME: ____________________ FIRST NAME: ____________________ ID (MATRICOLA): __________	Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.dis.uniroma1.it/~rosati/gd`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, ____________________

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$
\begin{array}{lcl}
A & \sqsubseteq & \neg F \\
B & \sqsubseteq & \neg F \\
B & \sqsubseteq & C \\
C & \sqsubseteq & D \sqcup E \\
D & \sqsubseteq & \exists R.A \\
E & \sqsubseteq & \exists R.B
\end{array}
$$

√

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) given the ABox $\mathcal{A} = \{C(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ is satisfiable (consistent), and if so, show a model for $\langle \mathcal{T}, \mathcal{A} \rangle$;

(c) given the ABox $\mathcal{A}' = \{C(a), \forall R.F(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A}' \rangle$ is satisfiable (consistent), and if so, show a model for $\langle \mathcal{T}, \mathcal{A}' \rangle$;

(d) given the ABox $\mathcal{A} = \{C(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $\exists R.\neg F(a)$, explaining your answer.

√ **Exercise 2** Given the following ASP program P:

```
r(x,y) :- p1(x), p2(y).
s(x,y) :- r(x,y), not p2(x), not p1(y).
t(x,y) :- s(x,y).
t(y,z) :- s(x,y), t(x,z).
v(x,y) :- r(x,y), not s(x,y).
w(x,y) :- r(x,y), not v(x,y).
p1(a). p1(b). p1(c). p2(b). p2(c). p2(d). p2(e).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

X

We want to formalize knowledge about the domain of students and professors. In particular, we want to formalize the following statements:

1. every student is a person;
2. every professor is a person;
3. student and professor are disjoint classes;
4. the property "is friend of" has domain person and range person;
5. the property "studies with" has domain student and range student;
6. the property "studies with" is a subproperty of the property "is friend of";
7. every professor that is the supervisor of at least one student is an active professor;
8. every professor that is also a student is a special professor;
9. every student studies with at least one student;
10. every student has a friend.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following: $\mathcal{ALC}$, Datalog with constraints, ASP, OWL, *DL-Lite*$_R$, $\mathcal{EL}$, *RL*, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

√

(a) Write an RDF/RDFS model representing the following statements about URIs `Person`, `Director`, `Actor`, `Writer`, `Movie`, `Country`, `Continent`, `Comedy`, `Drama`, `Man`, `Woman`, `filmedInYear`, `filmedInCountry`, `hasBoxOfficeGross`, `isDirectorOf`, `isWriterOf`, `actsIn`, `bornIn`, `Joe`, `Mary`, `Ann`, `Paul`, `Italy`, `France`, `Europe`, `ABC`, `XYZ`.

1. `Person`, `Director`, `Writer`, `Actor`, `Country`, `Continent`, `Movie`, `Comedy`, `Drama`, `Man`, and `Woman` are classes;
2. `Man` and `Woman` are subclasses of `Person`;

3. `Comedy` and `Drama` are subclasses of `Movie`;
4. `actsIn`, `bornIn`, `filmedInCountry`, `isDirectorOf` and `isWriterOf` are properties;
5. `isDirectorOf` has domain `Director` and range `Movie`;
6. `filmedInYear` has domain `Movie` and range `xsd:integer`;
7. `filmedInCountry` has domain `Movie` and range `Country`;
8. `bornIn` has domain `Person` and range `Country`;
9. `actsIn` has domain `Actor` and range `Movie`;
10. `isInContinent` has domain `Country` and range `Continent`;
11. Ann is the director and the writer of movie XYZ;
12. Joe and Paul act in movie ABC;
13. ABC was filmed in France in 2015;
14. Ann is a woman;
15. Italy and France are in Europe.

(b) Write SPARQL queries corresponding to the following requests: (b1) return all the directors of the movies filemd in Europe in 2016; (b2) return the dramas filmed in Italy and played by at least an Italian actor, and, optionally, the year when the movie was filmed.

Exercise 5

(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

(b) Add to the above ontology the axioms formalizing the following statements:

1. add a new property `isWrittenBy` and state that it is the inverse of `isWriterOf`;
2. add a new class `WrittenByMultipleAuthors` and state that it corresponds to the class of movies written by at least two writers;
3. add the new class `allFemaleCast` and state that such a class corresponds to the class consisting of every movie whose writers, directors and actors are all women;
4. add a new class `LowBudgetMovie` and state that it corresponds to the class of movies played by at most 5 actors;
5. add the new class `EuropeanMovie` and state that such a class corresponds to the class consisting of every movie having at least a director who was born in Europe.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning (that is, the models) of the ontology? if so, identify and list such axioms.

Exercise 6

(a) Axiomatize the following scenario, appropriately with action precondition and effect axioms, and obtain successor state axioms.

Fluents:

- `robotInsideRoom(s)` - The robot is inside the room in situation s.
- `windowOpen(s)` - The window is open in situation s.
- `robotCloseToWindow(s)` - The robot is close to the window in situation s.

Actions:

- `goInsideRoom` - The robot goes inside the room. This can be done if the robot is not inside the room, and has the effect that the robot will be inside the room.
- `goOutsideRoom` - The robot goes outside the room. This can be done if the robot is inside the room and the window is open, and has the effect that the robot will be outside the room.
- `goCloseToWindow` - The robot moves close to the window. This can be done if the robot is not close to the window and is inside the room, and has the effect that the robot will be close to the window.
- `closeWindow` - The robot closes the window. This can be done if the robot is close to the window and the window is open, and has the effect that the window will be closed.

Initial situation description: Initially the robot is outside the room, is not close to the window, and the window is open.

(b) Show, by applying regression, that the window is closed after the sequence of actions `goInsideRoom, goCloseToWindow, closeWindow, goOutsideRoom`, and that the sequence of actions is indeed executable.

(1a) Is T sat?

$$\begin{cases} \Delta^I = \{d\} & C^I = \emptyset \\ r^I = \emptyset & D^I = \emptyset \\ A^I = \emptyset & E^I = \emptyset \\ B^I = \emptyset & \end{cases}$$

This model satisfies all conditions

$$\begin{array}{lll} A & \sqsubseteq & \neg F \\ B & \sqsubseteq & \neg F \\ B & \sqsubseteq & C \\ C & \sqsubseteq & D \sqcup E \\ D & \sqsubseteq & \exists R.A \\ E & \sqsubseteq & \exists R.B \end{array}$$

(2a) ABox $\mathcal{A} = \{C(a)\}$ $\langle T, \mathcal{A} \rangle$ is sat?

$a^I = \{d\}$

$C^I = \{d\} \to (D \sqcup E)^I = \{d\} \to D^I = \{d\}$ 4 ✓

$\hookrightarrow (\exists R.A)^I = \{d\}$

$: \{d \in \Delta^I \mid \exists e \in \Delta^I \text{ s.t. } (d,e) \in r^I \text{ AND } e \in A^I\}$

$r^I = \{(d,e)\}$ $A^I = \{e\}$ 5 ✓

$\Delta^I = \{d, e\}$ $\hookrightarrow F^I = \{d\}$ 1 ✓ 2 ✓

$E^I = \{\}$ 6 ✓

$$I' = \begin{cases} \Delta = \{d,e\} & B = \{\ \} & E = \{\} \\ r = \{(d,e)\} & C = \{d\} & F = \{d\} \\ A = \{e\} & D = \{d\} & \end{cases}$$

Is I' a model of ABox A?

$a^I \in C^I \to \{d\} \in \{d\}$ ✓

Yes is sat!

(1c) ABox $\mathcal{A}' = \{C(a), \forall R.F(a)\}$

$a^I = \{d\}$

$\forall R.F = \{d, e\}$

$a^I \in C^I \to \{d\} \in \{d\}$ ✓

Is sat ✓

$a^I \in (\forall R.F)^I \to \{d\} \in \{d, e\}$ ✓

(1d) $A = \{C(a)\}$ $\langle T, A\rangle$ entails $\exists R.\neg F(a)$?

$A_0 = A \cup \{\neg\exists R.\neg F(a)\} = \{C(a), \forall R.F(a)\}$

$C_{GCI} = (\neg A \sqcup \neg F) \sqcap (\neg B \sqcup \neg F) \sqcap (\neg B \sqcup C) \sqcap (\neg C \sqcup D \sqcup E) \sqcap (\neg D \sqcup \exists R.A) \sqcap (\neg E \sqcup \exists R.B)$

C_{GCI} rule $= A_1 = A_0 \cup \{C_{GCI}(a)\}$

$\sqcap$ rule $A_2 = A_1 \cup \{(\neg A \sqcup \neg F)(a), (\neg B \sqcup \neg F)(a), (\neg B \sqcup C)(a), (\neg C \sqcup D \sqcup E)(a), (\neg D \sqcup \exists R.A)(a), (\neg E \sqcup \exists R.B)(a)\}$

$\sqcup$ rule $A_3 : A_2 \cup \{\neg C(a)\}$ CLASH | $A_4 : A_2 \cup \{D(a)\}$ | $A_5 : A_2 \cup \{E(a)\}$

$A_6 : A_4 \cup \{\neg D(a)\}$ CLASH | $A_7 : A_4 \cup \{\exists R.A(a)\}$

$A_8 : A_7 \cup \{R(a,b), A(b)\}$ new var, apply GCI-rule

$A_9 : A_8 \cup \{C_{GCI}(b)\}$

$A_{10} : A_9 \cup \{\neg A(b)\}$ CLASH | $A_{11} : A_9 \cup \{\neg F(b)\}$

$A_{12} : A_{11} \cup \{F(b)\}$ $(\forall R.F(a), R(a,b))$ CLASH

$A_{13} : A_5 \cup \{\neg E(a)\}$ CLASH | $A_{14} : A_5 \cup \{\exists R.B(a)\}$

$A_{15} : A_{14} \cup \{R(a,b), B(b)\}$ new var, apply C_{GCI}-rule

$A_{16} : A_{15} \cup \{C_{GCI}(b)\}$

$A_{17} : A_{16} \cup \{\neg B(b)\}$ CLASH | $A_{18} : A_{16} \cup \{\neg F(b)\}$

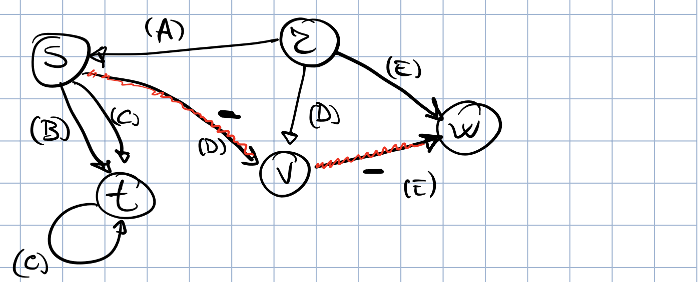
$A_{19} : A_{18} \cup \{F(b)\}$ $(\forall R.F(a), R(a,b))$ CLASH

All branch clash, KB entails $\exists R.\neg F(a)$

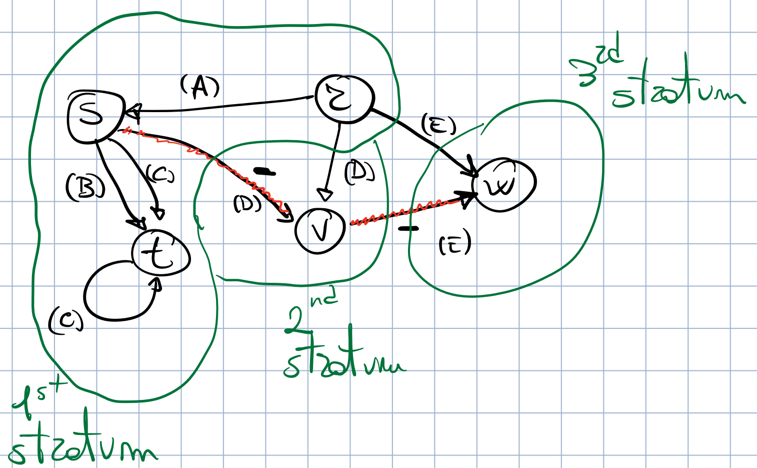
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

(2)

```
r(x,y) :- p1(x), p2(y).
s(x,y) :- r(x,y), not p2(x), not p1(y). (A)
t(x,y) :- s(x,y).
t(y,z) :- s(x,y), t(x,z).
v(x,y) :- r(x,y), not s(x,y).
w(x,y) :- r(x,y), not v(x,y).
p1(a). p1(b). p1(c). p2(b). p2(c). p2(d). p2(e).
```



No cycle with negative edges, so T is stratified.




```
r(x,y) :- p1(x), p2(y).
s(x,y) :- r(x,y), not p2(x), not p1(y). (A)
t(x,y) :- s(x,y).
t(y,z) :- s(x,y), t(x,z).
v(x,y) :- r(x,y), not s(x,y).
w(x,y) :- r(x,y), not v(x,y).
p1(a). p1(b). p1(c). p2(b). p2(c). p2(d). p2(e).
```

$$MM_0 = \{p_1(a), p_1(b), p_1(c), p_2(b), p_2(c), p_2(d), p_2(e)\}$$

$$S_1: \; T_{P_1}(MM_0) = \{r(a,b), r(a,c), r(a,d), r(a,e), r(b,b), r(b,c), r(b,d), r(b,e), r(c,b), r(c,c), r(c,d), r(c,e)\} = I_1$$

$$T_{P_1}(I_1 \cup MM_0) = \{s(a,d), s(a,e)\} = I_2$$

$$T_{P_1}(I_2 \cup MM_0) = \{t(a,d), t(a,e)\} = I_3$$

$$T_{P_1}(I_3 \cup MM_0) = \{t(d,d), t(e,d), t(d,e), t(e,e)\} = I_4$$

$$T_{P_1}(I_4 \cup MM_0) = I_4 \quad \text{fixpoint} \rightarrow MM_1 = MM_0 \cup I_4$$

$$S_2: \; T_{P_2}(MM_1) = \{v(a,b), v(a,c), v(b,b), v(b,c), v(b,d), v(b,e), v(c,b), v(c,c), v(c,d), v(c,e)\} = I_5$$

$$T_{P_2}(I_5 \cup MM_1) = I_5 \quad \text{fixpoint} \rightarrow MM_2 = MM_0 \cup I_5$$

$$S_3: \; T_{P_3}(MM_2) = \{w(a,d), w(a,e) = I_6$$

$$T_{P_3}(I_6 \cup MM_2) = I_6 \quad \text{fixpoint} \rightarrow MM_3 = MM_2 \cup I_6$$

MM_3 is the answer set of P.

④

```
@prefix rdf: <http://...> .
@prefix rdfs: <http://...> .
@prefix myns: <http://...> .

myns:Person rdf:type rdfs:class .
  " :Director  "  "
  " :Writer    "  "
  " :Actor     "  "
  " :Country   "  "
  " :Continent "  "
  " :Movie     "  "
  " :Comedy    "  "
  " :Drama     "  "
  " :Men       "  "
  " :Women     "  "
myns:Men rdfs:subClassOf myns:Person .
myns:Women rdfs:subClassOf myns:Person .
myns:Comedy rdfs:subClassOf myns:Movie .
myns:Drama rdfs:subClassOf myns:Movie .
myns:actsIn rdf:type rdf:Property .
  " :bornIn           "  "
  " :filmedInCountry  "  "
  " :isDirectorOf     "  "
  " :isWriterOf       "  "
myns:isDirectorOf rdfs:domain myns:Director .
myns:isDirectorOf rdfs:range myns:Movie .

myns:filmedInYear rdfs:domain myns:Movie .
myns:filmedInYear rdfs:range xsd:integer .
myns:filmedInCountry rdfs:domain myns:Movie .
myns:filmedInCountry rdfs:range myns:Country .
myns:bornIn rdfs:domain myns:Person .
myns:bornIn rdfs:range myns:Country .
myns:actsIn rdfs:domain myns:Actor .
myns:actsIn rdfs:range myns:Movie .
myns:isInContinent rdfs:domain myns:Country .
myns:isInContinent rdfs:range myns:Continent .

myns:Ann myns:isDirectorOf myns:XYZ .
myns:Ann myns:isWriterOf myns:XYZ .
myns:Joe myns:actsIn myns:ABC .
myns:Paul myns:actsIn myns:ABC .
myns:ABC myns:filmedInYear xsd:2015 .
myns:ABC myns:filmedInCounty myns:France .
myns:Ann rdf:type myns:Women .
myns:Italy myns:isInContinent myns:Europe .
myns:France myns:isInContinent myns:Europe .
```

b1:

```
PREFIX myns:<http://...>
SELECT ?x
WHERE { ?y myns:filmedInCounty ?z .
        ?z myns:isInContiment myns:Europe .
        ?y myns:filmedInYear xsd:2016 .
        ?x myns:isDirectorOf ?y . }
```

b2:

```
PREFIX
SELECT ?x ?y
WHERE { ?x rdf:type myns:Drama .
        ?x myns:filmedInCountry myns:Italy .
        ?z myns:actsIn ?x
        ?z myns:bornIn myns:Italy
        OPTIONAL { ?x myns:filmedInYear ?y . } }
```

5

RDFS -> OWL

a:

Declaration (Class(myns: Person))
Declaration (Class(myns: Director))
Declaration (Class(myns: Writer))
Declaration (Class(myns: Actor))
Declaration (Class(myns: Country))
Declaration (Class(myns: Continent))
Declaration (Class(myns: Movie))
Declaration (Class(myns: Comedy))
Declaration (Class(myns: Drama))
Declaration (Class(myns: Men))
Declaration (Class(myns: Women))

SubClassOf (myns: Men myns: Person)
SubClassOf (myns: Women myns: Person)
SubClassOf (myns: Comedy myns: Movie)
SubClassOf (myns: Drama myns: Movie)

Declaration (ObjectProperty(myns: actsIn))
Declaration (ObjectProperty(myns: bornIn))
Declaration (ObjectProperty(myns: filmedInCounty))
Declaration (ObjectProperty(myns: isDirectorOf))
Declaration (ObjectProperty(myns: isWriterOf))

ObjectPropertyDomain (myns:isDirectorOf myns:Director)
ObjectPropertyRange (myns:isDirectorOf myns:Movie)
ObjectPropertyDomain (myns: filmedInYear myns:Movie)
DataPropertyRange (myns: filmedInYear xsd:Integer)

ObjectPropertyDomain (myns: filmedInCountry myns:Movie)
ObjectPropertyRange (myns: filmedInCountry myns:Country)
ObjectPropertyDomain (myns:bornIn myns: Person)
ObjectPropertyRange (myns:bornIn myns:Country)
ObjectPropertyDomain (myns:actsIn myns: Actor)
ObjectPropertyRange (myns:actsIn myns:Movie)

ObjectPropertyDomain (myns:isInContinent myns:Country)
ObjectPropertyRange (myns:isInContinent myns:Continent)

ObjectPropertyAssertions (myns:isDirectorOf myns:Ann myns:X47)
ObjectPropertyAssertions (myns:isWriterOf myns:Ann myns:X47)
ObjectPropertyAssertions (myns:actsIn myns:Joe myns:X47)
ObjectPropertyAssertions (myns:actsIn myns:Paul myns:X47)
DataPropertyAssertions (myns:filmedInYear myns:ABC xsd:2015)
ObjectPropertyAssertions (myns:filmedInCounty myns:ABC myns:France)
ClassAssertion (myns: Women myns:Ann)

ObjectPropertyAssertions (myns:isInContinent myns:Italy myns:Europe)
ObjectPropertyAssertions (myns:isInContinent myns:France myns:Europe)

b:

Declaration (ObjectProperty(myns: isWrittenBy))
InverseObjectProperty(myns: isWrittenBy myns: isWriterOf)

Declaration (Class (myns: WrittenByMultipleAuthors))

3

1. every student is a person;
2. every professor is a person;
3. student and professor are disjoint classes;
4. the property "is friend of" has domain person and range person;
5. the property "studies with" has domain student and range student;
6. the property "studies with" is a subproperty of the property "is friend of";
7. every professor that is the supervisor of at least one student is an active professor;
8. every professor that is also a student is a special professor;
9. every student studies with at least one student;
10. every student has a friend.

	$\mathcal{ALC}$	Datalog	ASP	OWL	*DL-Lite*$_R$	$\mathcal{EL}$	*RL*	RDFS
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	—	X	X	X	—	X	—
4	X	X	X	X	X	X	—	X
5	X	X	X	X	X	X	—	X
6	—	X	X	X	X	—	X	X
7	X	—	—	X	—	X	X	—
8	X	—	—	X	—	X	X	—
9	X	—	—	X	—	X	—	—
10	X	—	—	X	X	X	—	—
	NO	NO	NO	YES	NO	NO	NO	NO

Knowledge Representation and Semantic Technologies – 13/6/2018

LAST NAME: ______________________

FIRST NAME: ______________________

ID (MATRICOLA): ____________

Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.dis.uniroma1.it/~rosati/gd`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede,

____

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$
\begin{array}{lcl}
A & \sqsubseteq & \neg F \\
B & \sqsubseteq & \neg F \\
B & \sqsubseteq & C \\
B & \sqsubseteq & \forall R.F \\
C & \sqsubseteq & E \\
D & \sqsubseteq & A \sqcup B \\
D & \sqsubseteq & \exists R.A \\
E & \sqsubseteq & \exists R.B
\end{array}
$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept B is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where B is satisfiable;

(c) given the ABox $\mathcal{A}' = \{D \sqcap F(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A}' \rangle$ is satisfiable (consistent), and if so, show a model for $\langle \mathcal{T}, \mathcal{A}' \rangle$;

(d) given the ABox $\mathcal{A}'' = \{D(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A}'' \rangle$ entails the assertion $A(a)$, explaining your answer.

Exercise 2 Given the following ASP program P:

```
r(x,y) :- p(x,y).
r(x,y) :- p(x,z), r(z,y).
s(x,y) :- r(x,y), not p(y,x).
t(x,y) :- r(x,y), not s(x,y).
v(x,y) :- s(x,y).
v(x,y) :- s(x,z), v(z,y).
w(x,y) :- v(x,y), not t(x,y).
p(a,b). p(b,c). p(c,d). p(c,e).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

We want to formalize knowledge about the domain of students and professors. In particular, we want to formalize the following statements:

1. every student is a person;
2. every professor is a person;
3. male and female are subclasses of person;
4. student and professor are disjoint classes;
5. the property "is friend of" has domain person and range person;
6. the property "studies with" has domain student and range student;
7. the property "studies with" is a subproperty of the property "is friend of";
8. every professor that is the supervisor of at least one student is an active professor;
9. every professor that is the supervisor of a male student and a female student is a very active professor;
10. every professor that is also a student is a special professor;
11. every professor is friend of at least one professor and is friend of at least one student.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following: $\mathcal{ALC}$, Datalog with constraints, ASP, OWL, *DL-Lite*$_R$, $\mathcal{EL}$, *RL*, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

see previous exam

(a) Write an RDF/RDFS model representing the following statements about URIs `Person`, `Director`, `Actor`, `Writer`, `Movie`, `Country`, `Continent`, `Comedy`, `Drama`, `Man`, `Woman`, `filmedInYear`, `filmedInCountry`, `hasBoxOfficeGross`, `isDirectorOf`, `isWriterOf`, `actsIn`, `bornIn`, `Joe`, `Mary`, `Ann`, `Paul`, `Italy`, `France`, `Europe`, `ABC`, `XYZ`.

1. Person, Director, Writer, Actor, Country, Continent, Movie, Comedy, Drama, Man, and Woman are classes;
2. Man and Woman are subclasses of Person;
3. Comedy and Drama are subclasses of Movie;
4. actsIn, bornIn, filmedInCountry, isDirectorOf and isWriterOf are properties;
5. isDirectorOf has domain Director and range Movie;
6. filmedInYear has domain Movie and range xsd:integer;
7. filmedInCountry has domain Movie and range Country;
8. bornIn has domain Person and range Country;
9. actsIn has domain Actor and range Movie;
10. isInContinent has domain Country and range Continent;
11. hasBoxOfficeGross has domain Movie and range xsd:integer;
12. Ann is the director and the writer of movie XYZ;
13. Joe and Paul act in movie ABC;
14. ABC was filmed in France in 2015;
15. Ann is a woman;
16. Italy and France are in Europe.

(b) Write SPARQL queries corresponding to the following requests: (b1) "return every director who directed at least one movie whose box office gross is above $10,000,000". (b2) return the directors of the movies filmed in Italy and played by at least an Italian actor, and, optionally, the box office gross of the movie and the writer of the movie.

Exercise 5

(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

(b) Add to the above ontology the axioms formalizing the following statements:

1. add a new property isWrittenBy and state that it is the inverse of isWriterOf;
2. add a new class WrittenByManyAuthors and state that it corresponds to the class of movies written by at least three writers;
3. add the new class allFemaleCast and state that such a class corresponds to the class consisting of every movie whose writers, directors and actors are all women;
4. add a new class LowBudgetMovie and state that it corresponds to the class of movies played by at most 5 actors;
5. add the new class ItalianMovie and state that such a class corresponds to the class consisting of every movie that was filmed in Italy.
6. add the new class Blockbuster and state that such a class corresponds to the class consisting of every movie whose box office gross is $100,000,000.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning (that is, the models) of the ontology? if so, identify and list such axioms.

Exercise 6

Axiomatize the following scenario, appropriately with action precondition and effect axioms, and obtain successor state axioms.

Fluents:

- doorOpen(s) - The door is open in situation s.
- windowOpen(s) - The window is open in situation s.
- robotColoseToDoor(s) - The robot is close to the door in situation s.
- robotColoseToWindow(s) - The robot is close to the window in situation s.

Actions:

- openDoor - The robot opens the door. This can be done if the robot is close to the door, and has the effect that the door will be open.
- closeDoor - The robot closes the door. This can be done if the robot is close to the door, and has the effect that the door will be closed.
- openWindow - The robot opens the window. This can be done if the robot is close to the window, and has the effect that the window will be open.
- closeWindow - The robot closes the window. This can be done if the robot is close to the window, and has the effect that the window will be closed.
- moveCloseToDoor - The robot moves close to the door. This can always be done, and has the effect that the robot will be close to the door.
- moveCloseToWindow - The robot moves close to the window. This can always be done, and has the effect that the robot will be close to the window.

Initial situation description: Initially the robot is not close to the door and not close to the window, the door is closed, and the window is closed.

A	$\sqsubseteq$	$\neg F$
B	$\sqsubseteq$	$\neg F$
B	$\sqsubseteq$	C
B	$\sqsubseteq$	$\forall R.F$
C	$\sqsubseteq$	E
D	$\sqsubseteq$	$A \sqcup B$
D	$\sqsubseteq$	$\exists R.A$
E	$\sqsubseteq$	$\exists R.B$

1 a

$$I = \begin{cases} A^I = B^I = C^I = D^I = E^I = \{\} \\ \Delta^I = \{d\} \\ r^I = \{\} \end{cases}$$

1 sat T

b B is sat? Let's modify I

$B^I = \{d\} \to \neg F = \{d\} \to F = \{e\}$ $\Delta = \{d, e\}$ $r = \{(d, e)\}$

$\to C = \{d\}$

$\to \forall R.F = \{x \in \Delta \mid y \in \Delta,$ IF $(x, y) \in r$ THEN $y \in F\}$

$= \{d, e\}$

$A = \{e\}$ ✓
$B = \{d\}$ ✓
$C = \{d\}$ ✓
$D = \{d\}$ ✓

$C = \{d\} \to E = \{d\}$

$D = \{d\} \to A \sqcup B = \{d, e\}$

$\to \exists R.A = \{x \in \Delta \mid y \in \Delta \ (x, y) \in r$ AND $y \in A\} \Rightarrow A = \{e\} \not\subseteq \neg F = \{d\}$

B is not sat wrt T

c ABox $\mathcal{A}' = \{D \sqcap F(a)\}$

$a^I = \{d\}$

$(D \sqcap F) = \{d\} \to D = \{d\}$ $\to \exists R.A = \{d\} \to A = \{e\}$ $r = \{(d, e)\}$

$\to A \sqcup B = \{d\} \to B = \{d\}$ but B cannot contain d

$F = \{d\}$ since $B \sqsubseteq \neg F$

The KB is unsat

d Tableau starts with

$A_0 = \mathcal{A} \cup \{\neg A(a)\} = \{D(a), \neg A(a)\}$

$C_{GCI} = (\neg A \sqcup \neg F) \sqcap (\neg B \sqcup \neg F) \sqcap (\neg B \sqcup C) \sqcap (\neg B \sqcup \forall R.F) \sqcap (\neg C \sqcup E) \sqcap (\neg D \sqcup A \sqcup B) \sqcap (\neg D \sqcup \exists R.A)$

$\sqcap (\neg E \sqcup \exists R.B)$

C_{GCI}-rule: $A_1 := A_0 \cup \{C_{GCI}(a)\}$

$\sqcap$-rule: $A_2 : A_1 \cup \{(\neg A \sqcup \neg F)(a), (\neg B \sqcup \neg F)(a), (\neg B \sqcup C)(a), (\neg B \sqcup \forall R.F)(a), (\neg C \sqcup E)(a), (\neg D \sqcup A \sqcup B)(a),$
$(\neg D \sqcup \exists R.A)(a), (\neg E \sqcup \exists R.B)(a)\}$

$\sqcup$-rule:

$A_3 : A_2 \cup \{\neg D(a)\}$ CLASH | $A_4 : A_2 \cup \{A(a)\}$ CLASH | $A_5 : A_2 \cup \{B(a)\}$

$A_6 : A_5 \cup \{\neg B(a)\}$ CLASH | $A_7 : A_5 \cup \{C(a)\}$

$A_8 : A_7 \cup \{\neg C(a)\}$ CLASH | $A_9 : A_7 \cup \{E(a)\}$

$A_{10} : A_9 \cup \{\neg E(a)\}$ CLASH | $A_{11} : A_9 \cup \{\exists R.B(a)\}$

$A_{12} : A_{11} \cup \{R(a,b), B(b)\}$

$A_{13} : A_{12} \cup \{\neg B(a)\}$ CLASH | $A_{14} : A_{12} \cup \{\forall R.F(a)\}$

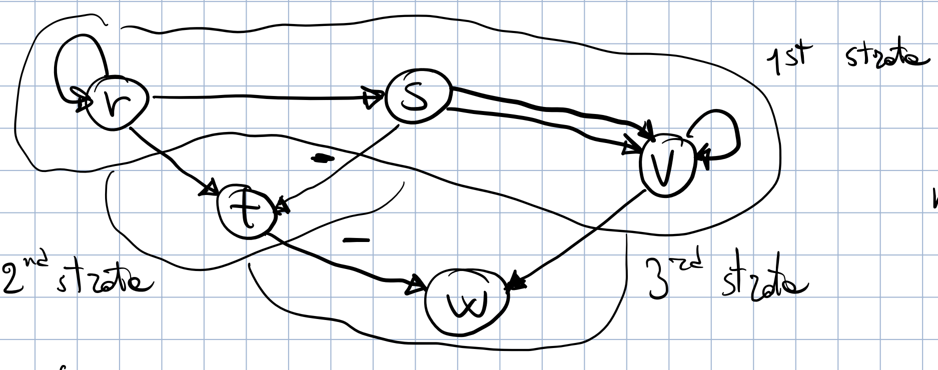
$A_{15} : A_{14} \cup \{F(b)\}$

C_{GCI}-rule: $A_{16} : A_{15} \cup \{C_{GCI}(b)\}$

$A_{16} : A_{15} \cup \{\neg B(b)\}$ CLASH | $A_{17} : A_{15} \cup \{\neg F(b)\}$ CLASH

$A(a)$ is entailed.

(2)



no cycle with - edges, P is stratified

$MM_0 = \{p(a,b), p(b,c), p(c,d), p(c,e)\}$

$S_1 : \; T_{P_1}(MM_0) = \{r(a,b), r(b,c), r(c,d), r(c,e)\} = I_1$

$T_{P_1}(I_1 \cup MM_0) = \{r(a,c), r(b,d), r(b,e)\} = I_2$

$T_{P_1}(I_2 \cup MM_0) = \{s(a,b), s(b,c), s(c,d), s(c,e), s(a,c), s(b,d), s(b,e)\} = I_3$

$T_{P_1}(I_3 \cup MM_0) = \{v(a,b), v(b,c), v(c,d), v(c,e), v(a,c), v(b,d), v(b,e)\} = I_4$

$T_{P_1}(I_4 \cup MM_0) = \{v(a,d), v(a,e)\} = I_5$ fixpoint

$MM_1 = I_5 \cup MM_0$

S_2: $T_{P_2}(MM_1) = \{\} = I_5$ fixpoint $\quad MM_2 =$

S_3: $T_{P_3}(MM_2) = \{w(a,b), w(b,c), w(c,d), w(c,e), w(a,c), w(b,d), w(b,e)$
$w(a,d), w(a,e)\} = I_6$ fixpoint

$MM_3 = I_6 \cup MM_2$

MM_3 is the answer set of P.

3

1. every student is a person;
2. every professor is a person;
3. male and female are subclasses of person;
4. student and professor are disjoint classes;
5. the property "is friend of" has domain person and range person;
6. the property "studies with" has domain student and range student;
7. the property "studies with" is a subproperty of the property "is friend of";
8. every professor that is the supervisor of at least one student is an active professor;
9. every professor that is the supervisor of a male student and a female student is a very active professor;
10. every professor that is also a student is a special professor;
11. every professor is friend of at least one professor and is friend of at least one student.

$\mathcal{ALC}$, Datalog, ASP, OWL, $DL\text{-}Lite_R$, $\mathcal{EL}$, RL, RDFS

	$\mathcal{ALC}$	Datalog	ASP	OWL	$DL\text{-}Lite_R$	$\mathcal{EL}$	RL	RDFS
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	—	X	X	—	—	X	X
5	X			X	—	X	—	X
6	X			X	—	X	—	X
7	—			X	X		X	X
8	X			X	—		—	
9	X			X				—
10	X			X	—	X	X	
11	X	—	—	X	—			
	NO	NO	NO	YES	NO	NO	NO	NO

Knowledge Representation and Semantic Technologies – 23/7/2018

LAST NAME: ______________________ FIRST NAME: ______________________ ID (MATRICOLA): __________	Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.dis.uniroma1.it/~rosati/gd`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, ____________________

√ **Exercise 1** Given the following $\mathcal{ALC}$ TBox:

$$\begin{array}{rcl} A & \sqsubseteq & \exists R.C \\ B & \sqsubseteq & \forall R.D \\ D & \sqsubseteq & \neg C \\ E & \sqsubseteq & A \sqcup \forall R.G \\ F & \sqsubseteq & B \sqcup \exists R.C \\ G & \sqsubseteq & D \end{array}$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept E is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where E is satisfiable;

(c) tell whether the concept $E \sqcap F$ is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where $E \sqcap F$ is satisfiable;

(d) given the ABox $\mathcal{A} = \{E(a), R(a, b)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $C(b)$, explaining your answer.

√ **Exercise 2** Given the following ASP program P:

```
r(x,y) :- p(x,y,z).
s(x,y) :- p(z,x,y).
t(x,y) :- r(x,y), s(y,z).
v1(x,y,z) :- r(x,y), s(y,z).
v2(x,y,z) :- v1(x,y,z), not p(x,y,z).
w(x,y) :- t(x,y), not s(x,y).
w(x,y) :- t(x,y), not r(x,y).
w(x,y) :- t(x,y), not v1(x,y,z).
w(x,y) :- t(x,y), not v2(x,y,z).
p(a,b,c). p(b,c,d). p(c,d,e). p(e,f,g).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

× We want to formalize knowledge about the domain of movies. In particular, we want to formalize the following statements:

1. every director is a person;
2. every actor is a person;
3. the property "worked in" has domain person and range movie;
4. the property "acted in" has domain actor and range movie;
5. the property "directed by" has domain movie and range director;
6. the property "acted in" is a subproperty of "worked in";
7. if a director was born in Italy, she/he is an Italian director;
8. if a director was born in a European country, she/he is a European director;
9. an actor-director is a director who is also an actor;
10. every movie directed by an actor-director is a special movie;
11. every movie directed by a European director is a European movie.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following ones: $\mathcal{ALC}$, Datalog, Datalog with constraints, ASP, OWL, $DL\text{-}Lite_R$, $\mathcal{EL}$, RL, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

√ (a) Write an RDF/RDFS model representing the following statements about URIs `Employee`, `Manager`, `Division`, `TopManager`, `Person`, `Man`, `Woman`, `City`, `livesIn`, `worksWith`, `isManagerOf`, `leadsDivision`, `locatedIn`, `Ann`, `Bob`, `Jane`, `Joe`, `Rome`, `Naples`, `ABC`, `XYZ`.

1. `Employee`, `Manager`, `TopManager`, `Division`, `Man`, `Woman` and `City` are classes;

2. `TopManager` is a subclass of `Manager` which is a subclass of `Employee`;
3. `worksWith`, `livesIn`, `isManagerOf`, `leadsDivision` and `locatedIn` are properties;
4. `isManagerOf` is a subproperty of `worksWith`;
5. `isManagerOf` has domain `Manager` and range `Employee`;
6. `worksWith` has domain `Employee` and range `Employee`;
7. `livesIn` has domain `Person` and range `City`;
8. `locatedIn` has domain `Division` and range `City`;
9. Jane is a manager;
10. Bob and Ann are employees;
11. Joe is manager of Bob;
12. Jane lives in Rome;
13. Mary leads division XYZ of the company;
14. division ABC is located in Naples.

√(b) Write SPARQL queries corresponding to the following requests: (b1) "return every employee that works with an employee that lives in Naples, and, optionally, the manager of such an employee"; (b2) "return every top manager that leads a division that is located in the same city where she/he lives".

Exercise 5

√(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

√(b) Add to the above ontology the axioms formalizing the following statements:

1. `City` and `Division` are disjoint classes;
2. every manager manages at least three employees;
3. every division is located in exactly one city;
4. every employee works with at least one man and works with at least one woman.
5. every manager leads at most one division;
6. ? `RomanManager` is the class of managers who lead a division located in Rome.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning of the ontology? if so, identify and list such axioms.

Exercise 6

Axiomatize the following scenario, appropriately with action precondition and effect axioms, and obtain successor state axioms.

Fluents:

- `windowOpen(s)` - The window is open in situation s.
- `robotCloseToWindow(s)` - The robot is close to the window in situation s.
- `objectInsideRoom(s)` - The object is inside the room in situation s.

Actions:

- `openWindow` - The robot opens the window. This can be done if the robot is close to the window, and has the effect that the window will be open.
- `closeWindow` - The robot closes the window. This can be done if the robot is close to the window, and has the effect that the window will be closed.
- `moveCloseToWindow` - The robot moves close to the window. This can always be done, and has the effect that the robot will be close to the window.
- `throwObject` - The robot throws the object outside the window. This requires that the oject is inside the room, the window is open and the robot is close to the window, and has the effect that the object will be outside the room.

Initial situation description: Initially the robot is not close to the window, the window is closed, and the object is inside the room.

A	$\sqsubseteq$	$\exists R.C$
B	$\sqsubseteq$	$\forall R.D$
D	$\sqsubseteq$	$\neg C$
E	$\sqsubseteq$	$A \sqcup \forall R.G$
F	$\sqsubseteq$	$B \sqcup \exists R.C$
G	$\sqsubseteq$	D

① a $\mathcal{I} = \begin{cases} A = B = D = E = F = G = \{\} \\ \Delta = \{d\} \quad r = \{\} \end{cases}$

$\mathcal{I}$ sat $\mathcal{T}$

b E sat $\mathcal{T}$? Modify $\mathcal{I}$

$E = \{d\} \rightarrow A \sqcup \forall R.G = \{d\}$

$\forall R.G : \{x \in \Delta \mid y \in \Delta \text{ IF } (x,y) \in r \text{ THEN } y \in G\} = \{d\}$

$\mathcal{I}' = \begin{cases} A = B = D = F = G = \{\} \quad E = \{d\} \\ \Delta = \{d\} \quad r = \{\} \end{cases}$ E sat $\mathcal{T}$

c $E \sqcap F$ sat $\mathcal{T}$? Modify $\mathcal{I}'$

$E \sqcap F = \{d\} \rightarrow E = \{d\} \checkmark$

$F = \{d\} \rightarrow (B \sqcup \exists R.C) = \{d\} \rightarrow B = \{d\} \rightarrow \forall R.D = \{d\}$

$\forall R.D : \{x \in \Delta \mid y \in \Delta \text{ IF } (x,y) \in r \text{ THEN } y \in D\} = \{d\}$

$E \sqcap F$ sat $\mathcal{T}$ ✓

d Tableau starts with

$A_0 : A \cup \{\neg C(b)\} = \{E(a), R(a,b), \neg C(b)\}$

$C_{GCI} : \{(\neg A \sqcup \exists R.C) \sqcap (\neg B \sqcup \forall R.D) \sqcap (\neg D \sqcup \neg C) \sqcap (\neg E \sqcup A \sqcup \forall R.G) \sqcap (\neg F \sqcup B \sqcup \exists R.C) \sqcap (\neg G \sqcup D)\}$

C_{GCI}-rule $A_1 : A_0 \cup \{C_{GCI}(a)\}$

$\sqcap$-rule: $A_2 : A_1 \cup \{(\neg A \sqcup \exists R.C)(a), (\neg B \sqcup \forall R.D)(a), (\neg D \sqcup \neg C)(a), \underline{(\neg E \sqcup A \sqcup \forall R.G)(a)}, (\neg F \sqcup B \sqcup \exists R.C)(a), (\neg G \sqcup D)(a)\}$

$A_3 : A_2 \cup \{\neg \underline{E(a)}\}$ CLASH

$A_4 : A_2 \cup \{A(a)\}$

$A_5 : A_2 \cup \{\forall R.G(a)\}$

$A_6 : A_4 \cup \{\neg \underline{A(a)}\}$ CLASH

$A_7 : A_4 \cup \{\exists R.C(a)\}$

$A_8: A_7 \cup \{R(a,c), C(c)\}$ apply $C_{G\sqcap}$-rule on c

$A_9: A_8 \cup \{C_{G\sqcap}(c)\}$

$A_{10}: A_9 \cup \{\neg D(c)\}$ | $A_{11}: A_9 \cup \{\neg C(c)\}$ CLASH

$A_{12}: A_{10} \cup \{\neg G(c)\}$ | $A_{13}: A_{10} \cup \{D(c)\}$ CLASH

$A_{13}: A_{12} \cup \{\neg B(c)\}$ | $A_{14}: A_{12} \cup \{\forall R.D(c)\}$

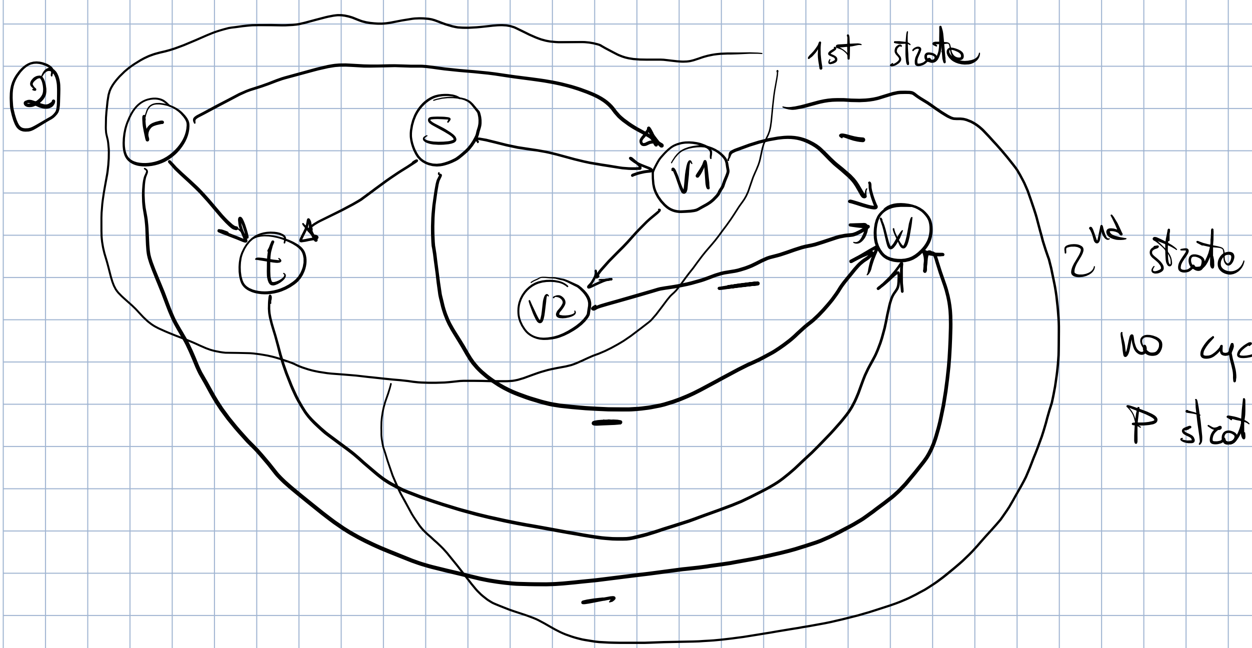
OPEN | $A_{15}: A_{14} \cup \{D(c)\}$ CLASH

branch open

KB doesn't entail C(b)

---

(2)



no cycle with - edges

P stratified

$MM_0 = \{p(a,b,c), p(b,c,d), p(c,d,e), p(e,f,g)\}$

$S_1: T_{P_1}(MM_0) = \{r(a,b), r(b,c), r(c,d), r(e,f), s(b,c), s(c,d), s(d,e), s(f,g)\} = I_1$

$T_{P_1}(I_1 \cup MM_0) = \{t(a,b), t(b,c), t(c,d), t(e,f), v_1(a,b,c), v_1(b,c,d), v_1(c,d,e), v_1(e,f,g)\} = I_2$

$T_{P_1}(I_2 \cup MM_0) = \{\} = I_2$ fixpoint $\rightarrow MM_1 = I_2 \cup MM_0$

$S_2: T_{P_2}(MM_1) = \{w(a,b), w(e,f), w(b,c), w(c,d)\} = I_3$ fixpoint $\rightarrow MM_2 = I_3 \cup MM_1$

MM_2 is the answer set of P

4

a:

```
myns:Employee      rdf:type            rdfs:Class .
myns:Manager       rdf:type            rdfs:Class .
myns:TopManager    rdf:type            rdfs:Class .
myns:Division      rdf:type            rdfs:Class .
myns:Men           rdf:type            rdfs:Class .
myns:Women         rdf:type            rdfs:Class .
myns:City          rdf:type            rdfs:Class .
myns:TopManager    rdfs:subClassOf     myns:Manager .
myns:Manager       rdfs:subClassOf     myns:Employee .
myns:worksWith     rdf:type            rdf:Property .
myns:livesIn       rdf:type            rdf:Property .
myns:isManagerOf   rdf:type            rdf:Property .
myns:leadsDivision rdf:type            rdf:Property .
myns:locatedIn     rdf:type            rdf:Property .
myns:isManagerOf   rdfs:subPropertyOf  myns:worksWith .
myns:isManagerOf   rdfs:domain         myns:Manager .
myns:isManagerOf   rdfs:range          myns:Employee .
myns:worksWith     rdfs:domain         myns:Employee .
myns:worksWith     rdfs:range          myns:Employee .
myns:livesIn       rdfs:domain         myns:Person .
myns:livesIn       rdfs:range          myns:City .
myns:locatedIn     rdfs:domain         myns:Division .
myns:locatedIn     rdfs:range          myns:City .
myns:Jane          rdf:type            myns:Manager .
myns:Bob           rdf:type            myns:Employee .
myns:Ann           rdf:type            myns:Employee .
myns:Joe           myns:isManagerOf    myns:Bob .
myns:Jane          myns:livesIn        myns:Rome .
myns:Mary          myns:leadsDivision  myns:XYZ .
myns:ABC           myns:locatedIn      myns:Naples .
```

b1:

```
PREFIX myns: <http://...>
SELECT ?x ?y
WHERE { ?x myns:worksWith ?z .
        ?z myns:livesIn myns:Naples .
        OPTIONAL { ?y myns:isManagerOf ?x . } }
```

b2:

```
PREFIX rdf: <http://...> myns: <http://...>
SELECT ?x
WHERE { ?x rdf:type myns:TopManager .
        ?x myns:leadsDivision ?y .
        ?x myns:livesIn ?z .
        ?y myns:locatedIn ?z . }
```

5

a:

```
Declaration(Class(myns:Employee))
Declaration(Class(myns:Manager))
Declaration(Class(myns:TopManager))
Declaration(Class(myns:Division))
Declaration(Class(myns:Men))
Declaration(Class(myns:Women))
Declaration(Class(myns:City))
SubClassOf(myns:TopManager myns:Manager)
SubClassOf(myns:Manager myns:Employee)
Declaration(ObjectProperty(myns:worksWith))
Declaration(ObjectProperty(myns:livesIn))
Declaration(ObjectProperty(myns:isManagerOf))
Declaration(ObjectProperty(myns:leadsDivision))
Declaration(ObjectProperty(myns:locatedIn))
SubPropertyOf(myns:isManagerOf myns:worksWith)
ObjectPropertyDomain(myns:isManagerOf myns:Manager)
ObjectPropertyRange(myns:isManagerOf myns:Employee)
ObjectPropertyDomain(myns:worksWith myns:Employee)
ObjectPropertyRange(myns:worksWith myns:Employee)
ObjectPropertyDomain(myns:livesIn myns:Person)
ObjectPropertyRange(myns:livesIn myns:City)
ObjectPropertyDomain(myns:locatedIn myns:Division)
ObjectPropertyRange(myns:locatedIn myns:City)
ClassAssertion(myns:Manager myns:Jane)
ClassAssertion(myns:Employee myns:Bob)
ClassAssertion(myns:Employee myns:Ann)
ObjectPropertyAssertion(myns:isManagerOf myns:Joe myns:Bob)
ObjectPropertyAssertion(myns:livesIn myns:Jane myns:Rome)
ObjectPropertyAssertion(myns:leadsDivision myns:Mary myns:XYZ)
ObjectPropertyAssertion(myns:locatedIn myns:ABC myns:Naples)
```

b:

- DisjointClasses(myns:City myns:Division)
- SubClassOf(
    myns:Manager
    ObjectMinCardinality(3 myns:isManagerOf myns:Manager))
- SubClassOf(
    myns:Division
    ObjectExactCardinality(1 myns:locatedIn myns:City))
- SubClassOf(
    myns:Employee
    ObjectIntersectionOf(
      ObjectMinCardinality(1 myns:worksWith myns:Men)
      ObjectMinCardinality(1 myns:worksWith myns:Women)
    ))

- SubClassOf(
  myns:Manager
  ObjectMaxCardinality(1 myns:leadsDivision myns:Division))

- SubClassOf(myns:RomanManager
  ObjectIntersectionOf(myns:Manager
  ObjectSomeValuesFrom(myns:leadsDivision myns:Division)
  ObjectSomeValuesFrom(myns:locatedIn myns:Rome)))

(3)

1. every director is a person;
2. every actor is a person;
3. the property "worked in" has domain person and range movie;
4. the property "acted in" has domain actor and range movie;
5. the property "directed by" has domain movie and range director;
6. the property "acted in" is a subproperty of "worked in";
7. if a director was born in Italy, she/he is an Italian director;
8. if a director was born in a European country, she/he is a European director;
9. an actor-director is a director who is also an actor;
10. every movie directed by an actor-director is a special movie;
11. every movie directed by a European director is a European movie.

$\mathrm{DIR.} \sqcap \mathrm{born.ITALY} \sqsubseteq \mathrm{ITAL.DIR.}$

$\mathrm{MOVIE} \sqcap \exists \mathrm{directedby.ACTORDIR} \sqsubseteq \mathrm{SPEC.MOVIE}$

	$\mathcal{ALC}$	Datalog	ASP	OWL	$DL\text{-}Lite_R$	$\mathcal{EL}$	RL	RDFS
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	—	—	X	X	—	—	X
4	X	—	—	X	X	—	—	X
5	X			X	X	—	—	X
6	—			X	X	X	X	X
7	X			X	—	X	X	—
8	X			X	—	X	X	—
9	X			X	—	X	X	—
10	X			X	—	X	X	—
11	X			X	—	X	X	—

Knowledge Representation and Semantic Technologies – 23/1/2019

LAST NAME: ____________________ FIRST NAME: ____________________ ID (MATRICOLA): __________	Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.dis.uniroma1.it/~rosati/gd`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, ____________________

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$\begin{array}{rcl} A & \sqsubseteq & \neg F \\ B & \sqsubseteq & C \sqcap G \\ C & \sqsubseteq & F \sqcup \exists R.G \\ D & \sqsubseteq & E \sqcap F \\ E & \sqsubseteq & \exists R.A \\ F & \sqsubseteq & \forall R.B \\ G & \sqsubseteq & \forall R.\neg G \end{array}$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept B is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where B is satisfiable;

(c) tell whether the concept D is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where D is satisfiable;

(d) given the ABox $\mathcal{A} = \{G(a), R(a,b)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $G(b)$, explaining your answer.

Exercise 2 Given the following ASP program P:

```
r(x,y) :- p(x,y,z).
s(y,z) :- p(x,y,z).
t(x,z) :- p(x,y,z).
u(x,y,z) :- r(x,y), s(y,z).
v1(x,y,z) :- u(x,y,z), not t(x,z).
v2(x,y,z) :- u(x,y,z), not r(x,y).
w(x,y) :- u(x,y,z), not v1(x,y,z).
w(x,z) :- w(x,y), w(z,y).
p(a,b,c). p(d,e,f). p(f,b,d).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

We want to formalize knowledge about the domain of movies. In particular, we want to formalize the following statements:

1. every director is a person;
2. every actor is a person;
3. the property "worked in" has domain person and range movie;
4. the property "acted in" has domain actor and range movie;
5. the property "directed by" has domain movie and range director;
6. the property "acted in" is a subproperty of "worked in";
7. if a director was born in a European country, she/he is a European director;
8. an actor-director is a director who is also an actor;
9. every movie directed by an actor-director is a special movie;
10. every movie directed by a European director is a European movie.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following ones: $\mathcal{ALC}$, Datalog, Datalog with constraints, ASP, OWL, *DL-Lite*$_R$, $\mathcal{EL}$, *RL*, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

(a) Write an RDF/RDFS model representing the following statements about URIs `Employee`, `Manager`, `Division`, `TopManager`, `Person`, `Man`, `Woman`, `City`, `livesIn`, `worksWith`, `isManagerOf`, `leadsDivision`, `worksInDivision`, `locatedIn`, `Ann`, `Bob`, `Jane`, `Joe`, `Rome`, `Naples`, `Milan`, `ABC`, `XYZ`.

See previous exam

1. `Employee`, `Manager`, `TopManager`, `Division`, `Man`, `Woman` and `City` are classes;
2. `TopManager` is a subclass of `Manager` which is a subclass of `Employee`;

3. `worksWith`, `livesIn`, `isManagerOf`, `leadsDivision` and `locatedIn` are properties;
4. `isManagerOf` is a subproperty of `worksWith`;
5. `isManagerOf` has domain `Manager` and range `Employee`;
6. `worksInDivision` has domain `Employee` and range `Division`;
7. `worksWith` has domain `Employee` and range `Employee`;
8. `livesIn` has domain `Person` and range `City`;
9. `locatedIn` has domain `Division` and range `City`;
10. Jane is a manager;
11. Bob and Ann are employees;
12. Joe is manager of Bob;
13. Jane lives in Rome;
14. Mary leads division XYZ of the company;
15. division ABC is located in Milan.

(b) Write SPARQL queries corresponding to the following requests: (b1) return all the managers of the male employees that live in Milan; (b2) return the cities of the divisions for which at least a female employee works; (b3) return every manager that works in a division located in Naples, and, optionally, the city where the manager lives.

Exercise 5

(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

(b) Add to the above ontology the axioms formalizing the following statements:

1. every manager leads at most one division;
2. `City` and `Division` are disjoint classes;
3. a `SpecialDivision` is a division for which at least four managers work;
4. a `SpecialManager` is a manager that manages at least two male employees and at least two female employees;
5. a `RomanEmployee` is an employee who works in a division located in Rome and lives in Rome.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning of the ontology? if so, identify and list such axioms.

Exercise 6

Axiomatize the following scenario, appropriately with action precondition and effect axioms, and obtain successor state axioms.

Fluents:

- `doorOpen(s)` - The door is open in situation s.
- `robotCloseToDoor(s)` - The robot is close to the door in situation s.
- `objectCloseToDoor(s)` - The object is close to the door in situation s.

Actions:

- `openDoor` - The robot opens the door. This can be done if the robot is close to the door and the door is closed, and has the effect that the door will be open.
- `closeDoor` - The robot closes the door. This can be done if the robot is close to the door and the door is open, and has the effect that the door will be closed.
- `moveCloseToDoor` - The robot moves close to the door. This can always be done, and has the effect that the robot will be close to the door.
- `moveObject` - The robot pushes the object out of the room. This requires that the object is close to the door, the robot is close to the door and the door is open, and has the effect that the object will be not close to the door.

Initial situation description: Initially the robot is not close to the door, the door is closed, and the object is close to the door.

SubClassOf(myns:SpecialDivision
ObjectSomeValuesFrom(
ObjectMinCardinality(2 myns:workInDivision myns:Manager)
myns:Division))

?

SubClassOf(myns:SpecialDivision
ObjectIntersectionOf(myns:Division
ObjectMinCardinality(2
ObjectInverseOf(myns:workInDivision)
myns:Manager)))

1a $A = B = C = D = E = F = G = \emptyset$

$\Delta = \{d\} \quad R = \{\ \}$ ✓

$$\begin{array}{lcl} A & \sqsubseteq & \neg F \\ B & \sqsubseteq & C \sqcap G \\ C & \sqsubseteq & F \sqcup \exists R.G \\ D & \sqsubseteq & E \sqcap F \\ E & \sqsubseteq & \exists R.A \\ F & \sqsubseteq & \forall R.B \\ G & \sqsubseteq & \forall R.\neg G \end{array}$$

1b $B = \{d\} \rightarrow C \sqcap G = \{d\} \rightarrow C = \{d\} \rightarrow F \sqcup \exists R.G = \{d\} \rightarrow F = \{d\}$

$G = \{d\} \rightarrow \forall R.\neg G = \{d\}$

$\downarrow$

$\{x \in \Delta \mid y \in \Delta \text{ IF } (x,y) \in R \text{ THEN } y \in \neg G\}$

$= \{d\}$

$A = D = E = \emptyset \quad B = C = F = G = \{d\}$

$\Delta = \{d\} \quad R = \{\ \}$ ✓

$\Delta = \{d, e\}$

1c $D = \{d\} \rightarrow E \sqcap F = \{d\} \rightarrow E = \{d\} \rightarrow \exists R.A = \{d\} \rightarrow R = \{(d,e)\} \quad A = \{e\}$

$F = \{d\} \rightarrow \forall R.B = \{d, e\} \quad B = \{e\}$

$$B \sqsubseteq C \sqcap G \sqsubseteq F \sqcup \underbrace{\exists R.G \sqcap \forall R.\neg G}_{\emptyset}$$

$$D \sqsubseteq E \sqcap F \sqsubseteq \exists R.A \sqcap \forall R.B \sqsubseteq \exists R.\neg F \sqcap \forall R.(F \sqcup \underbrace{\exists R.G \sqcap \forall R.\neg G}_{\emptyset})$$

$$\sqsubseteq \underbrace{\exists R.\neg F \sqcap \forall R.F}_{\emptyset}$$

✗

$$\begin{array}{lcl} A & \sqsubseteq & \neg F \\ B & \sqsubseteq & C \sqcap G \\ C & \sqsubseteq & F \sqcup \exists R.G \\ D & \sqsubseteq & E \sqcap F \\ E & \sqsubseteq & \exists R.A \\ F & \sqsubseteq & \forall R.B \\ G & \sqsubseteq & \forall R.\neg G \end{array}$$

d Tableau starts with:

$\mathcal{A}_0 = \mathcal{A} \cup \{\neg G(b)\} = \{G(a), R(a,b), \neg G(b)\}$

$C_{GCI} = (\neg A \sqcup \neg F) \sqcap (\neg B \sqcup (C \sqcap G)) \sqcap (\neg C \sqcup F \sqcup \exists R.G) \sqcap (\neg D \sqcup (E \sqcap F)) \sqcap (\neg E \sqcup \exists R.A) \sqcap$
$(\neg F \sqcup \forall R.B) \sqcap (\neg G \sqcup \forall R.\neg G)$

$C_{GCI\text{-rule}} = \{C_{GCI}(a)\}$

$\sqcap$-rule: $\mathcal{A}_1 = \mathcal{A}_0 \cup \{(\neg A \sqcup \neg F)(a), (\neg B \sqcup (C \sqcap G))(a), (\neg C \sqcup F \sqcup \exists R.G)(a), \underline{(\neg D \sqcup (E \sqcap F))(a)},$
$(\neg E \sqcup \exists R.A)(a), (\neg F \sqcup \forall R.B)(a), (\neg G \sqcup \forall R.\neg G)(a)\}$

$A_2 : A_1 \cup \{\neg D(a)\}$

$A_{12} : A_2 \cup \{\underline{\neg G(a)}\}$ CLASH

$A_{13} : A_2 \cup \{\forall R. \neg G(a)\}$

$A_{14} : A_{13} \cup \{\neg F(a)\}$

⋮

$A_{15} : A_{13} \cup \{\forall R.B(a)\}$

$A_{16} : A_{13} \cup \{B(b)\}$

C_{GCI}-rule : $A_{17} : A_{16} \cup \{C_{GCI}(b)\}$

$A_{18} : A_{17} \cup \{\underline{\neg B(b)}\}$ CLASH

$A_{19} : A_{17} \cup \{(C \sqcap G)(b)\}$

$A_{20} : A_{19} \cup \{C(b), \underline{G(b)}\}$ CLASH

$A_3 : A_1 \cup \{(E \sqcap F)(a)\}$

$A_4 : A_3 \cup \{E(a), F(a)\}$

$A_5 : A_4 \cup \{\underline{\neg F(a)}\}$ CLASH

$A_6 : A_4 \cup \{\forall R.B(a)\}$

$A_7 : A_6 \cup \{B(b)\}$

C_{GCI}-rule : $A_8 : A_7 \cup \{C_{GCI}(b)\}$

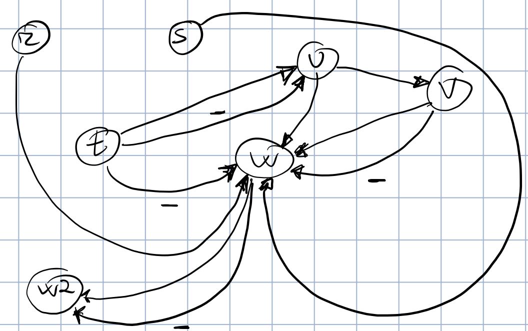
$A_9 : A_8 \cup \{\underline{\neg B(b)}\}$ CLASH

$A_{10} : A_8 \cup \{(C \sqcap G)(b)\}$

$A_{11} : A_{10} \cup \{C(b), \underline{G(b)}\}$ CLASH

branch open if I continue $\langle T, A\rangle$ doesn't entail $G(b)$

2)



r, s, t 1st strata no cycles with - edges

P stratified

u, v 2nd strata

w 3rd strata

w2 4th strata

S_1: $MM_0 = \{p(a,b,c), p(d,e,f), p(f,b,d)\}$

$T_{P_1}(MM_0) = \{r(a,b), r(d,e), r(f,b), s(b,c), s(e,f), s(b,d), t(a,c), t(d,f), t(f,d)\} = I_1$

$MM_1 = I_1 \cup MM_0$

S_2: $T_{P_2}(MM_1) = \{u(a,c)\} = I_2$

$T_{P_2}(I_2 \cup MM_1) = \{v(c,a)\} = I_3$

$w(y,z) : r(x,y), v(x,z), \neg v(y,z)$

$$T_{P_2}(I_3 \cup MM_1) = \{ w(b,e), w(b,c) \} = I_4$$

$$MM_2 = I_4 \cup MM_1$$

S_3: $T_{P_3}(MM_2) = \{ \} = I_4$ MM_2 is the answer sets of P ✓

4b

```
PREFIX ~~ ~~
SELECT ?x
WHERE { ?x myns:isManagerOf ?y .
        ?y rdf:type myns:Male .
        ?y myns:livesIn myns:Milan. }
```

```
PREFIX ~~ ~~
SELECT ?x
WHERE { ?y myns:locatedIn ?x .
        ?z myns:worksInDivision ?y .
        ?z rdf:type myns:Male. }
```

```
PREFIX ~~ ~~
SELECT ?x ?y
WHERE { ?x rdf:type myns:Manager .
        ?x myns:worksInDivision ?z .
        ?z myns:locatedIn myns:Naples.
        OPTIONAL { ?x myns:livesIn ?y . } }
```

3

1. every director is a person;
2. every actor is a person;
3. the property "worked in" has domain person and range movie;
4. the property "acted in" has domain actor and range movie;
5. the property "directed by" has domain movie and range director;
6. the property "acted in" is a subproperty of "worked in";
7. if a director was born in a European country, she/he is a European director;
8. an actor-director is a director who is also an actor;
9. every movie directed by an actor-director is a special movie;
10. every movie directed by a European director is a European movie.

	$\mathcal{ALC}$	Datalog	ASP	OWL	$DL\text{-}Lite_R$	$\mathcal{EL}$	RL	RDFS
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	—	—	X	X	X	—	X
4	X	—	—	X	X	X	—	X
5	X	—	—	X	X	X	—	X
6	—	X	X	X	X	—	X	X
7	X	—	—	X	—	X	?	—
8	X	X	X	X	—	X	X	—
9	X	—	—	X	—	X	?	—
10	X	—	—	X	—	X		—
	NO	NO	NO	YES	NO	NO	NO	NO

Knowledge Representation and Semantic Technologies – 18/2/2019

LAST NAME: ____________________ FIRST NAME: ____________________ ID (MATRICOLA): __________	Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.dis.uniroma1.it/~rosati/gd`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, ____________________

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$\begin{array}{rcl} A & \sqsubseteq & \neg F \\ B \sqcap F & \sqsubseteq & \exists R.H \\ C & \sqsubseteq & D \sqcup E \\ D & \sqsubseteq & F \sqcap \forall R.A \\ E & \sqsubseteq & \exists R.G \\ H & \sqsubseteq & F \end{array}$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept C is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where C is satisfiable;

(c) tell whether the concept $B \sqcap D$ is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where $B \sqcap D$ is satisfiable;

(d) given the ABox $\mathcal{A} = \{B(a), C(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $E(a)$, explaining your answer.

Exercise 2 Given the following ASP program P:

```
r(x,y) :- p(x,y,z).
s(y,z) :- p(x,y,z).
t(x,z) :- p(x,y,z).
u(x,y) :- t(x,y), not t(y,x).
v(x,y) :- u(y,x).
w(x,z) :- s(x,y), v(y,z), not t(x,z).
w(y,z) :- r(x,y), u(x,z), not v(y,z).
w2(x,y,z) :- w(x,y), w(y,z), not w(x,z).
p(a,b,c). p(d,e,f). p(f,b,d).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

We want to formalize knowledge about the domain of movies. In particular, we want to formalize the following statements:

1. every director is a person;
2. every actor is a person;
3. the property "worked in" has domain person and range movie;
4. the property "acted in" has domain actor and range movie;
5. the property "directed by" has domain movie and range director;
6. the property "acted in" is a subproperty of "worked in";
7. a famous director is a director who directed more than 20 movies;
8. if a director was born in Italy, she/he is an Italian director;
9. an actor-director is a director who is also an actor;
10. every movie directed by an actor-director is a special movie;
11. every movie directed by an Italian director is an Italian movie.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following ones: $\mathcal{ALC}$, Datalog, Datalog with constraints, ASP, OWL, *DL-Lite*$_R$, $\mathcal{EL}$, *RL*, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

See previous exam

(a) Write an RDF/RDFS model representing the following statements about URIs `Employee`, `Manager`, `Division`, `TopManager`, `Person`, `Man`, `Woman`, `City`, `livesIn`, `worksWith`, `isManagerOf`, `leadsDivision`, `worksInDivision`, `locatedIn`, `Ann`, `Bob`, `Jane`, `Joe`, `Rome`, `Naples`, `Milan`, `ABC`, `XYZ`.

1. `Employee`, `Manager`, `TopManager`, `Division`, `Man`, `Woman` and `City` are classes;
2. `TopManager` is a subclass of `Manager` which is a subclass of `Employee`;

3. `worksWith`, `livesIn`, `isManagerOf`, `leadsDivision` and `locatedIn` are properties;
4. `isManagerOf` is a subproperty of `worksWith`;
5. `isManagerOf` has domain `Manager` and range `Employee`;
6. `worksInDivision` has domain `Employee` and range `Division`;
7. `worksWith` has domain `Employee` and range `Employee`;
8. `livesIn` has domain `Person` and range `City`;
9. `locatedIn` has domain `Division` and range `City`;
10. Jane is a manager;
11. Bob and Ann are employees;
12. Joe is manager of Bob;
13. Jane lives in Rome;
14. Mary leads division XYZ of the company;
15. division ABC is located in Milan.

(b) Write SPARQL queries corresponding to the following requests: (b1) return all the female managers of the male employees that live in Naples; (b2) return the cities of the divisions for which at least a male manager works; (b3) return every manager that works in a division located in Rome, and, optionally, the city where the manager lives.

Exercise 5

(a) Write an OWL ontology that formalizes the domain described at point (a) of Exercise 4.

(b) Add to the above ontology the axioms formalizing the following statements:

1. every manager leads at most one division;
2. `City` and `Division` are disjoint classes;
3. a `SpecialDivision` is a division for which at least four managers work;
4. a `SpecialManager` is a manager that manages at least two male employees and at least two female employees;
5. every employee lives in exactly one city;
6. every male employee works with at least one male employee and works with at least one female employee.

Then, tell whether the resulting OWL ontology is redundant, i.e.: can some of the axioms constituting the ontology be deleted without changing the meaning of the ontology? if so, identify and list such axioms.

Exercise 6

Axiomatize the following scenario, appropriately with action precondition and effect axioms, and obtain successor state axioms.

Fluents:

- `doorClosed(s)` - The door is closed in situation s.
- `robotCloseToDoor(s)` - The robot is close to the door in situation s.
- `objectNotCloseToDoor(s)` - The object is not close to the door in situation s.

Actions:

- `openDoor` - The robot opens the door. This can be done if the robot is close to the door and the door is closed, and has the effect that the door will be open.
- `closeDoor` - The robot closes the door. This can be done if the robot is close to the door and the door is open, and has the effect that the door will be closed.
- `moveCloseToDoor` - The robot moves close to the door. This can always be done, and has the effect that the robot will be close to the door.
- `moveObject` - The robot pushes the object out of the room. This requires that the object is close to the door, the robot is close to the door and the door is open, and has the effect that the object will be not close to the door.

Initial situation description: Initially the robot is not close to the door, the door is closed, and the object is close to the door.

$$
\begin{aligned}
A &\sqsubseteq \neg F\\
B \sqcap F &\sqsubseteq \exists R.H\\
C &\sqsubseteq D \sqcup E\\
D &\sqsubseteq F \sqcap \forall R.A\\
E &\sqsubseteq \exists R.G\\
H &\sqsubseteq F
\end{aligned}
$$

1 a: $A = (B \sqcap F) = C = D = E = H = \{\}$

$\Delta = \{d\}$ $\quad r = \{\}$ $\quad$ I sat T

b: $C = \{d\} \rightarrow D \sqcup E = \{d\} \rightarrow E = \{d\}$

$E = \{d\} \rightarrow \exists R.G = \{d\} \rightarrow G = \{e\}$ $\quad \Delta = \{d, e\}$ $\quad r = \{(d, e)\}$

$A = (B \sqcap F) = D = H = \{\}$ $\quad C = E = \{d\}$

$\Delta = \{d, e\}$ $\quad r = \{(d, e)\}$ $\quad$ I' sat T, C is sat wrt T

c: $B \sqcap D = \{d\}$

$B = \{d\}$

$D = \{d\} \rightarrow F \sqcap \forall R.A = \{d\}$

$F = \{d\}$

$\forall R.A = \{x \in \Delta \mid y \in \Delta, \text{ IF } (x, y) \in r \text{ THEN } y \in A\}$ $\quad A = \{e\}$

$A = \{e\} \sqsubseteq \neg F = \{e\}$ ✓ $\quad = \{d, e\}$

$B \sqcap F = \{d\} \sqsubseteq \exists R.H = \{d\}$ $\quad (\quad)$

$\hookrightarrow H = \{e\}$ but $H \not\sqsubseteq F$

So $B \sqcap D$ is not sat wrt T

d: Tableau starts with $\mathcal{A}_0 = \{B(a), C(a), \neg E(a)\}$

$C_{GCI}: (\neg A \sqcup \neg F) \sqcap (\neg B \sqcup \neg F \sqcup \exists R.H) \sqcap (\neg C \sqcup D \sqcup E) \sqcap (\neg D \sqcup F \sqcap \forall R.A) \sqcap (\neg E \sqcup \exists R.G)$

$\sqcap (\neg H \sqcup F)$

C_{GCI}-rule $\quad A_1 : A_0 \cup \{C_{GCI}(a)\}$

$\sqcap$-rule: $A_2 : A_1 \cup \{(\neg A \sqcup \neg F)(a), (\neg B \sqcup \neg F \sqcup \exists R.H)(a), (\neg C \sqcup D \sqcup E)(a), (\neg D \sqcup F \sqcap \forall R.A)(a) \sqcap (\neg E \sqcup \exists R.G)(a)$

$(\neg H \sqcup F)(a)\}$

$\sqcup$-rule: $A_3 : A_2 \cup \{\neg C(a)\}$ CLASH | $A_4 : A_2 \cup \{D(a)\}$ | $A_5 : A_2 \cup \{E(a)\}$ CLASH

$A_5: A_4 \cup \{\neg D(a)\}$ CLASH

$A_6: A_4 \cup \{F \sqcap \forall R.A(a)\}$

$A_7: A_6 \cup \{F(a), \forall R.A(a)\}$

$A_8: A_7 \cup \{\neg E(a)\}$ CLASH | $A_9: A_7 \cup \{\exists R.G(a)\}$

$A_{10}: A_9 \cup \{R(a,b), G(b)\}$

$A_{11}: A_{10} \cup \{A(b)\}$

$A_{12}: A_{11} \cup \{\neg A(a)\}$ | $A_{13}: A_{11} \cup \{\neg F(a)\}$ CLASH

$A_{13}: A_{12} \cup \{\neg B(a)\}$ CLASH | $A_{14}: A_{12} \cup \{\neg F(a)\}$ CLASH | $A_{15}: A_{12} \cup \{\exists R.H(a)\}$

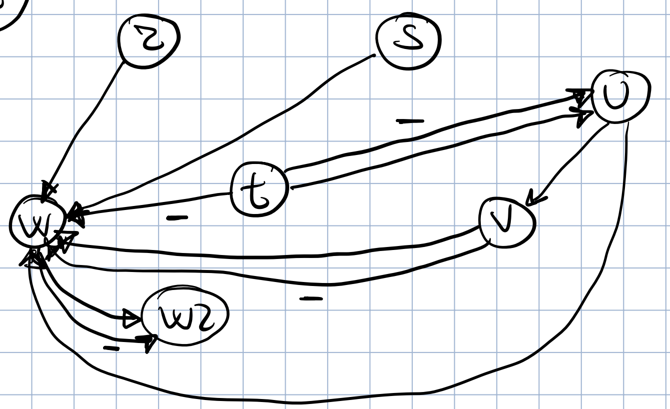
$A_{16}: A_{15} \cup \{R(a,c)\, H(c)\}$

$A_{17}: A_{16} \cup \{\neg H(a)\}$ OPEN | $A_{18}: A_{16} \cup \{F(a)\}$ CLASH

branch open

$\langle T, A \rangle$ doesn't entail $E(a)$

②



no cycle with − edges, P stratified

1st strata r, s, t

2nd strata u, v

3rd strata w

4th strata w2

$MM_0 = \{p(a,b,c), p(d,e,f), p(f,b,d)\}$

$T_{P_1}(MM_0) = \{r(a,b), r(d,e), r(f,b), s(b,c), s(e,f), s(b,d), t(a,c), t(d,f), t(f,d)\}$
$= I_1$ fixpoint $MM_1 = MM_0 \cup I_1$

$T_{P_2}(MM_1) = \{u(a,c)\} = I_2$

$T_{P_2}(I_2 \cup MM_1) = \{v(c,a)\} = I_3$ fixpoint $MM_2 = MM_1 \cup I_3$

$T_{P_3}(MM_2) = \{w(b,a), w(b,c)\} = I_4$ fixpoint $MM_3 = MM_2 \cup I_4$

$T_{P_3}(MM_3) = \{\} = I_4$ MM_3 is the answer set of P.

4b :

```
PREFIX ~ ~
SELECT ?x
WHERE { ?x rdf:type myns:Women .
        ?x myns:isManagerOf ?y .
        ?y rdf:type myns:Man .
        ?y myns:livesIn myns:Naples . }
```

```
PREFIX ~ ~
SELECT ?x
WHERE { ?y myns:worksInDivision ?z .
        ?y rdf:type myns:Man .
        ?z myns:locatedIn ?x . }
```

```
PREFIX ~ ~
SELECT ?x ?y
WHERE { ?x myns:isManagerOf ?z .
        ?x myns:worksInDivision ?a .
        ?a myns:locatedIn myns:Rome .
        OPTIONAL { ?x myns:livesIn ?y . }}
```

3

1. every director is a person;
2. every actor is a person;
3. the property "worked in" has domain person and range movie;
4. the property "acted in" has domain actor and range movie;
5. the property "directed by" has domain movie and range director;
6. the property "acted in" is a subproperty of "worked in";
7. a famous director is a director who directed more than 20 movies;
8. if a director was born in Italy, she/he is an Italian director;
9. an actor-director is a director who is also an actor;
10. every movie directed by an actor-director is a special movie;
11. every movie directed by an Italian director is an Italian movie.

	$\mathcal{ALC}$	Datalog	ASP	OWL	*DL-Lite*$_R$	$\mathcal{EL}$	*RL*	RDFS
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	—	X
4	X	X	X	X	X	X	—	X
5	X	X	X	X	X	X	—	X
6	—	X	X	X	X	X	X	X
7	X	—	—	X	—	—	—	—
8	X	X	X	X	—	X	X	—
9	X	X	X	X	—	X	X	—
10	X	—	—	X	—	X	X	—
11	X	—	—	X	—	X	X	—
	NO	NO	NO	YES	NO	NO	NO	NO

Knowledge Representation and Semantic Technologies – 14/9/2021

LAST NAME: ____________ FIRST NAME: ____________ ID (MATRICOLA): ____________	Autorizzo la pubblicazione del mio voto di questo esame sul sito web `http://www.diag.uniroma1.it/rosati/krst`, secondo quanto prevede il decreto legislativo 196/2003 (codice in materia di protezione dei dati personali) che dichiaro di conoscere. In fede, ____________

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$
\begin{array}{rcl}
A & \sqsubseteq & C \\
D & \sqsubseteq & \exists R.C \\
E & \sqsubseteq & \forall R.F \\
E & \sqsubseteq & B \\
F & \sqsubseteq & \neg B \\
G \sqcap B & \sqsubseteq & \exists R.A \\
H & \sqsubseteq & G \\
H & \sqsubseteq & \exists R.B
\end{array}
$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept $E \sqcap G$ is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where $E \sqcap G$ is satisfiable;

(c) tell whether the concept $E \sqcap H$ is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where $E \sqcap H$ is satisfiable;

(d) given the ABox $\mathcal{A} = \{E(a), R(a,b)\}$, use the tableau method to establish whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $F(b)$.

Exercise 2 Given the following ASP program P:

```
r(x,y) :- p(x,y,v).
s(x,y) :- p(v,x,y).
t(x,z) :- r(x,y), s(y,z), not r(x,z).
t(x,z) :- t(x,y), t(y,z), not r(x,z).
u(x,y) :- t(x,y), not s(x,y).
v(x,y) :- u(y,x).
w(x,z) :- r(x,y), s(y,z), not u(x,z).
w(x,y) :- t(x,y), not v(x,y).
w(x,y) :- v(x,y), not t(x,y).
p(a,b,c). p(c,d,e). p(e,f,f).
```

(a) tell whether P is stratified;

(b) compute the answer sets of P.

Exercise 3

We want to formalize knowledge about persons and kinship relationships. In particular, we want to formalize the following statements:

1. every person has a father and has a mother;
2. fathers are male and mothers are female;
3. the father of a father is a grandfather;
4. the mother of a mother is a grandmother;
5. the brother of a parent is an uncle.

(a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following ones: $\mathcal{ALC}$, Datalog, Datalog with constraints, ASP, OWL, *DL-Lite*$_R$, $\mathcal{EL}$, *RL*, RDFS, motivating your choice;

(b) express the above knowledge in the formalism chosen at the previous point.

Exercise 4

(a) Write an RDF/RDFS model representing the following statements about URIs `Employee`, `Manager`, `Division`, `TopManager`, `Person`, `Man`, `Woman`, `City`, `livesIn`, `worksWith`, `isManagerOf`, `leadsDivision`, `worksInDivision`, `locatedIn`, `Ann`, `Bob`, `Jane`, `Joe`, `Rome`, `Naples`, `Milan`, `ABC`, `XYZ`.

1. `Employee`, `Manager`, `TopManager`, `Division`, `Man`, `Woman` and `City` are classes;
2. `TopManager` is a subclass of `Manager` which is a subclass of `Employee`;
3. `worksWith`, `livesIn`, `isManagerOf`, `leadsDivision` and `locatedIn` are properties;
4. `isManagerOf` is a subproperty of `worksWith`;
5. `isManagerOf` has domain `Manager` and range `Employee`;

see past exam

6. both `worksInDivision` and `leadsDivision` have domain `Employee` and range `Division`;
7. `worksWith` has domain `Employee` and range `Employee`;
8. `livesIn` has domain `Person` and range `City`;
9. `locatedIn` has domain `Division` and range `City`;
10. Jane is a manager;
11. Bob and Ann are employees;
12. Joe is manager of Bob;
13. Jane lives in Rome;
14. Mary leads division XYZ of the company;
15. division ABC is located in Milan.

(b) Write SPARQL queries corresponding to the following requests: (b1) return every employee that works in a division located in Naples, and, optionally, the manager of such an employee; (b2) return every division that is located in the city where the manager that leads the division lives in; (b3) return every top manager that leads a division for which Ann works, and, optionally, the city where the manager lives.

Exercise 5

Given the RL knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$ is the following TBox:

$$\begin{array}{l} hasMother \sqsubseteq hasParent \\ hasFather \sqsubseteq hasParent \\ hasParent^- \sqsubseteq hasChild \\ \exists hasChild.\top \sqsubseteq PARENT \\ \exists hasChild.PARENT \sqsubseteq GRANDPARENT \\ \exists hasChild.GRANDPARENT \sqsubseteq GREAT\text{-}GRANDPARENT \end{array}$$

and $\mathcal{A}$ is the following ABox:

$$\begin{array}{lll} hasMother(John, Ann), & hasFather(John, Bob), & hasMother(Ann, Mary), \\ hasFather(Ann, Paul), & hasMother(Bob, Jane), & hasMother(Paul, Liz) \end{array}$$

1. compute the materialization of the ABox $\mathcal{A}$ with respect to the TBox $\mathcal{T}$;
2. tell whether the concept assertion $GREAT\text{-}GRANDPARENT(Liz)$ is entailed by $\langle \mathcal{T}, \mathcal{A} \rangle$.

(5) $hasMother \sqsubseteq hasParent$ $\qquad$ $hasM(Jo, A)$ $\quad$ $hasM(A, M)$

$hasFather \sqsubseteq hasParent$ $\qquad$ $hasM(B, J)$ $\quad$ $hasM(P, L)$

$hasParent^- \sqsubseteq hasChild$ $\qquad$ $hasF(Jo, B)$ $\quad$ $hasF(A, P)$

$\exists hasChild.\top \sqsubseteq PARENT$

$\exists hasChild.PARENT \sqsubseteq GRANDPARENT$

$\exists hasChild.GRANDPARENT \sqsubseteq GREAT\text{-}GRANDPARENT$

- $hasP(Jo, A)$ $\quad$ $hasP(A, M)$ $\quad$ $hasP(B, J)$ $\quad$ $hasP(P, L)$ $\quad$ $hasP(Jo, B)$ $\quad$ $hasP(A, P)$

$hasC(A, Jo)$ $\quad$ $hasC(M, A)$ $\quad$ $hasC(J, B)$ $\quad$ $hasC(L, P)$ $\quad$ $hasC(B, Jo)$ $\quad$ $hasC(P, A)$

$P(A)$ $\quad$ $P(M)$ $\quad$ $P(J)$ $\quad$ $P(L)$ $\quad$ $P(B)$ $\quad$ $P(P)$

$GP(M)$ $\quad$ $GP(J)$ $\quad$ $GP(L)$ $\quad$ $GP(P)$

$GGP(L)$

Liz
↓
Paul
↓
Ann
↓
John

✓

(1) a) $A = D = E = F = G \sqcap B = H = \{\}$

$\Delta = \{d\}$ $\qquad$ $\mathcal{I} = \{\}$ $\qquad$ T is sat

A	$\sqsubseteq$	C
D	$\sqsubseteq$	$\exists R.C$
E	$\sqsubseteq$	$\forall R.F$
E	$\sqsubseteq$	B
F	$\sqsubseteq$	$\neg B$
$G \sqcap B$	$\sqsubseteq$	$\exists R.A$
H	$\sqsubseteq$	G
H	$\sqsubseteq$	$\exists R.B$

b) $E \sqcap G = \{d\} \rightarrow E = \{d\}$ $\qquad$ $G = \{d\}$

$\hookrightarrow B = \{d\}$

$\hookrightarrow \forall R.F = \{x \in \Delta \mid y \in \Delta \text{ IF } (x,y) \in \mathcal{I} \text{ THEN } y \in F\} = \{d, e\}$

$G \sqcap B = \{d\} \rightarrow \exists R.A = \{d\}$ $\quad$ $A = \{e\} \rightarrow C = \{e\}$

$\Delta = \{d, e\}$

$\mathcal{I} = \{(d, e)\}$

$F = \{e\}$

1 ✓ $\quad$ 5 ✓
2 ✓ $\quad$ 6 ✓
3 ✓ $\quad$ 7 ✓
4 ✓ $\quad$ 8

$H = \{\}$

$A = C = F = \{e\}$ $\quad$ $B = D = E = G = \{d\}$ $\quad$ $\Delta = \{d, e\}$ $\quad$ $\mathcal{I} = \{(d, e)\}$

$E \sqcap G$ sat wrt T ✓

c) $E \sqcap H \sqsubseteq \forall R.F \sqcap \exists R.B \sqsubseteq \forall R.\neg B \sqcap \exists R.B = \{\}$

$E \sqcap H$ is not sat wrt $\mathcal{T}$

d) Tableau starts with : $A_0 = \{E(a), R(a,b), \neg F(b)\}$

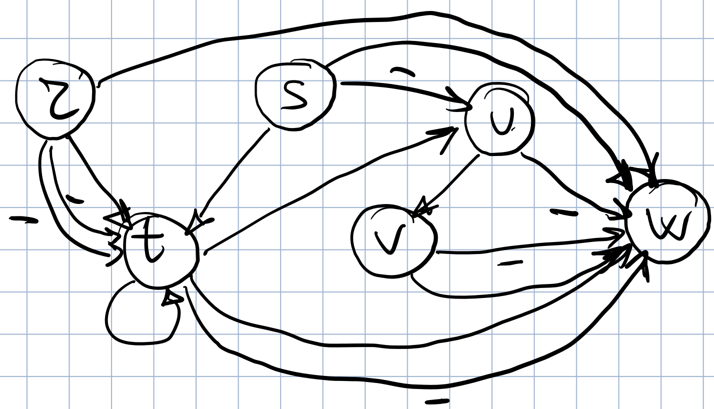
GCI-rule; $A_1 = A_0 \cup \{(\neg A \sqcup C)(a), (\neg D \sqcup \exists R.C)(a), (\underline{\neg E \sqcup \forall R.F})(a), (\neg E \sqcup B)(a), (\neg F \sqcup \neg B)(a),$
$\sqcup$-rule
$(\neg G \sqcup \neg B \sqcup \exists R.A)(a), (\neg H \sqcup G)(a), (\neg H \sqcup \exists R.B)(a)\}$

$A_2 : A_1 \cup \{\underline{\neg E(a)}\}$
CLASH

$A_3 : A_1 \cup \{\forall R.F(a)\}$
$A_4 : A_3 \cup \{\underline{F(b)}\}$
CLASH

KB entails $F(b)$

(2)



no cycles with - edges
P stratified

r, s 1^{st} strata
t, u, v 2^{nd} strata
w 3^{rd} strata

$MM_0 = \{p(a,b,c), p(c,d,e), p(e,f,g)\}$

S_1: $T_{P_1}(MM_0) = \{r(a,b), r(c,d), r(e,f), s(b,c), s(d,e), s(f,g)\} = I_1$

$MM_1 = I_1 \cup MM_0$

S_2: $T_{P_2}(MM_1) = \{t(a,c), t(c,e), t(e,g)\} = I_2$

$T_{P_2}(I_2 \cup MM_1) = \{t(a,e), t(c,g)\} = I_3$

$T_{P_2}(I_3 \cup MM_1) = \{u(a,e), u(c,g)\} = I_4$

$T_{P_2}(I_4 \cup MM_1) = \{v(e,a), v(g,c)\} = I_5$ $\quad MM_2 = I_5 \cup MM_1$

$T_{P_3}(MM_2) = \{w(a,c), w(c,e), w(e,g)\} = I_6$

$T_{P_3}(I_6 \cup MM_2) = \{w(a,e), w(c,g)\} = I_7$

$MM_3 = I_7 \cup MM_2$

MM_3 is the answer sets of P

(45):

```
PREFIX ~ ~
SELECT ?x ?y
WHERE { ?x myns:worksInDivision ?a .
        ?a myns:locatedIn myns:Naples .
        OPTIONAL { ?y myns:isManagerOf ?x . }}
```

```
PREFIX ~ ~
SELECT ?x
WHERE { ?z myns:leadsDivision ?x .
        ?z myns:livesIn ?y .
        ?x myns:locatedIn ?y . }
```

```
PREFIX ~ ~
SELECT ?x ?y
WHERE { ?x rdf:type myns:TopManager .
        ?x myns:leadsDivision ?z .
        myns:Ann myns:worksInDivision ?z .
        OPTIONAL { ?x myns:livesIn ?y . }}
```

(3)

1. every person has a father and has a mother; PERSON $\sqsubseteq \exists$hasFather.$\top$
2. fathers are male and mothers are female; FATHER $\sqsubseteq$ MALE
3. the father of a father is a grandfather;
4. the mother of a mother is a grandmother;
5. the brother of a parent is an uncle.

$\mathcal{ALC}$, Datalog, ASP, OWL, $DL\text{-}Lite_R$, $\mathcal{EL}$, RL, RDFS

	$\mathcal{ALC}$	Datalog	ASP	OWL	$DL\text{-}Lite_R$	$\mathcal{EL}$	RL	RDFS
1	X	—	—	X	X	X	—	—
2	X	X	X	X	X	X	X	X
3	X	X	X	X	—	X	X	
4	X			X	—	X	X	
5	X			X	—	X	X	