21/7/2017 EX3

domenica 3 gennaio 2021 11:03

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. 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: ASP, Datalog with constraints, ALC, OWL, $DL\text{-}Lite_{\mathcal{R}}$, EL, RDFS, motivating your choice;

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

a)

1) All languages admitt it

2) All languages admitt it

3) Not allow in EL because it cannot have negation, even if we are considering distinct classes, Not Allow in Datalog with constraint because it does not admitt negation because Datalog with constraint refers to positive datalog, Negation is also Not allow in RDFS

4) Not allow in EL because it doesn't admitt the unqualified existential restriction, ALC is ok because we can rewrite inverse role with $\forall$ constructor

5) Same as previous

6) Not admitt in ALC because ALC doesn't have subroles, also not allow in EL because it doesn't have subroles

7) Not allow in RDFS because we don't have the conjunction, Not allow in DL-Lite for the same reason and also because here we have qualified existential restriction that are not allow in DL-Lite

8) Not allow in RDFS because of the presence of conjuction and the same is for DL-Lite$_{\mathcal{R}}$

9) Not allow in RL because we have Concept expression in the right side of the formula and in RL we admit only atomic concept. Also in ASP and Datalog with constraint this is not admitt because we violate the safeness condition. Not allow in DL-Lite because we have qualified existential restriction, also for RDFS.

10) Same as previous

b) the only clake is OWL

1) subClassOf(myNS: Student myNS: Person)
2) subClassOf(myNS: Profesor myNS: Person)
3) DisJointClasses(myNS: Profesor myNS: Student)
4) subClassOf(ObjectSomeValueFrom(ObjectInverseOf(myNS: isFriendOf) owl:Thing) myNS: Person)
   subClassOf(ObjectSomeValueFrom(myNS: isFriendOf owl:Thing) myNS: Person)
5) subClassOf(ObjectSomeValueFrom(ObjectInverseOf(myNS: studiesWith) owl:Thing) myNS: STudent)
   subClassOf(ObjectSomeValueFrom(myNS: studiesWith owl:Thing) myNS Student)
6) subObjectPropaty OF (myNS: studiesWith myNS: isFriendOf)
7) subClassOf (ObjectIntersectionOF(myNS: Profesor
   ObjectSomeValuesFrom(myNS: isSuperVisorOf myNS: STudent))
   myNS: Active Profesor)
8) subClassOf(ObjectIntersectionOf(myNS: Profesor
   myNS: Student)
   myNS: Special Profesor)

9) subClassOf(myNS: Student ObjectSomeValuesFrom(myNS studiesWith myNS: STudent))
10) subClassOf(myNS: Student ObjectSomeValuesFrom(ObjectInverse(myNS: isFriendOF)
   myNS: Person))