

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. active professor is a subclass of professor;
 4. student and professor are disjoint classes;
 5. the property "is friend of" has domain person and range person;
 6. the property "is supervisor of" has domain professor and range student;
 7. the property "studies with" has domain student and range student;
 8. the property "studies with" is a subproperty of the property "is friend of";
 9. every professor that is the supervisor of at least one student is an active professor;
 10. every professor that is also a student is a special professor.
- (a) Choose the most appropriate knowledge representation language for expressing the above knowledge among the following: *ACC*, *Datalog*, *ASP*, *OWL*, *DL-Lite_R*, *EL*, *RL*, *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) All languages admitt it
- 4) Not possible in EL because there is no negation, No possible in RDFS because of the negation.
- 5) Not possible in EL because there is no inverse role.
- 6) Same as previous
- 7) Same as previous
- 8) Not possible in ALC because there is no role hierarchy, no possible in EL for the same reason
- 9) Not possible in RDFS because of conjunction, not possible in DL-Lite because we cannot do conjunction and qualified existential restriction.
- 10) Not possible in RDFS because of conjunction, not possible in DL-Lite because we cannot do conjunction

b) RL because is better for reasoning task than other languages that we can use

- 1) STUDENT subseq PERSON
- 2) PROFESSOR subseq PERSON
- 3) ACTIVE-PROFESSOR subseq PROFESSOR
- 4) STUDENT and PROFESSOR subseq bottom
- 5) Exists isFriendOf. T subseq PERSON
Exists isFriendOf⁻. T subseq PERSON
- 6) Exists isSupervisorOf. T subseq PROFESSOR
Exists isSupervisorOf⁻. T subseq STUDENT
- 7) Exists studiesWith. T subseq STUDENT

Exists studiesWith⁻. T subseteq STUDENT

8) studiesWith subseteq isFriendOf

9) PROFESSOR and Exist isSupervisor. STUDENT subseteq ACTIVE-PROFESSOR

10) PROFESSOR and STUDENT subseteq SPECIAL-PROFESSOR