Knowledge Representation and Semantic Technologies – 14/7/2014

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:

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

(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) given the ABox $\mathcal{A} = \{A(a)\}$, tell whether the concept assertion $\exists R.E(a)$ is entailed by $\langle \mathcal{T}, \mathcal{A} \rangle$, 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(x,y).
v(x,y) :- r(x,y), not s(x,y).
p(a,b). p(b,c). p(c,d). p(d,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 employees and managers. In particular, we want to formalize the following statements:

1. every employee is a person;
2. every manager is an employee;
3. property "is manager of" has domain manager and range employee;
4. property "is manager of" is a subproperty of property "works with";
5. "works with" is a symmetric property, that is, if e_1 works with e_2 then e_2 works with e_1;
6. "works with" is a transitive property, that is, if e_1 works with e_2 and e_2 works with e_3 then e_1 works with e_3.

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

(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`, `Country`, `Movie`, `filmedIn`, `isDirectorOf`, `actsIn`, `Ann`, `Bob`, `Jane`, `Paul`, `Italy`, `ABC`.

1. `Person`, `Director`, `Actor`, `Country` and `Movie` are classes;
2. `Director` and `Actor` are subclasses of `Person`;
3. `actsIn`, `filmedIn` and `isDirectorOf` are properties;
4. `isDirectorOf` has domain `Director` and range `Movie`;
5. `flmedIn` has domain `Movie` and range `Country`;
6. `actsIn` has domain `Actor` and range `Movie`;
7. Bob and Jane are actors;
8. ABC is a movie;
9. ABC was filmed in Italy;
10. Ann is the director of ABC.

(b) Write SPARQL queries corresponding to the following requests: (b1) "return every movie whose director also acts in that movie"; (b2) "return every actor that acts in a movie directed by Ann or by Paul".

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 class `MovieDirectedByActor` and state that it corresponds to the class of movies directed by actors;
2. add a new property `isDirectedBy` and state that it is the inverse of `isDirectorOf`;
3. every movie is directed by at least one director;
4. `Movie` and `Person` are disjoint classes;
5. every movie is filmed in at most one country.

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:

- `hungry(s)` - The turkey is hungry in situation s.
- `foodAvailable(s)` - The food is available to the turkey in situation s.

Actions:

- `findFood` - The turkey finds some food. This can be done if the food is currently not available to the turkey, and has the effect that the food will be available.
- `eat` - The turkey eats the food. This requires that the food is available, and has the effect that the turkey will be not hungry and the food will not be available anymore.
- `wait` - A no-op; it has no effect on any fluent, and can always be performed.

Initial situation description: Initially the turkey is hungry and the food is not available.

(b) Show, by applying regression, that the turkey will be not hungry after the sequence of actions wait, findFood, wait, eat, and that the sequence of actions is indeed executable.