Assignment 2 – Ontologies / Description Logic

Due: Monday, February 7th (in class, after class)

Problem 1. (10+4+3 Points)

Consider the following (oversimplified!) description logic ontology (TBox):

i. Organism ≡ Animal ⊔ Plant
ii. Person ⊑ Animal
iii. Grass ⊑ Plant
iv. Cow ⊑ Animal ⊓ ∀eats.Grass
v. Carnivore ≡ Organism ⊓ ∀eats.Animal
vi. Rancher ≡ Person ⊓ ∀eats.Cow ⊓ ∃owns.Ranch

a) Translate the above description logic (DL) axioms into first-order predicate logic (FO) formulas. Hint: To translate the concept expressions on the left-hand-side and right-hand-side of the above axioms, use the translations t_x and t_y given in class. To translate an equivalence C ≡ D or a concept inclusion C ⊑ D, compute t_x for the lhs and rhs, respectively, and use

- $\forall x (t_x(C) \leftrightarrow t_x(D))$ for the equivalence or
- $\forall x (t_x(C) \rightarrow t_x(D))$ for the implication.

b) When unfolding a concept expressions say E, we can replace a concept C (occurring in E) by an equivalent concept D, i.e., for which C ≡ D holds. If C ⊑ D holds, we can also replace C by D but need to remember that the resulting expression E′ is no longer equivalent to E.

"Unfold" the expression E = Person ⊓ ∀eats.Cow ⊓ ∃owns.Ranch (equivalent to Ranchers in the above ontology) until it contains only base concepts. Note that the resulting expression E′ might not be equivalent to E (e.g., if one replaces Grass by Plant in a conjunction, then a possibly larger result is obtained).

c) In the above ontology, what is the relation between Rancher and Carnivore? For example, is every Rancher a Carnivore? How about the other way round? Explain.

Problem 2 (1+2+3 Points).

a) What is the difference between a TBox and an ABox, i.e., what kind of information is stored in either one?
b) What is the difference between evaluating a query and reasoning with a query (or with two queries)? Which problem is harder in general?
c) What is the relation between evaluating a formula (*val* mapping on the slides) in logic and running a query? Say what corresponds to what (e.g., A in logic corresponds to X in databases, B in logic corresponds to Y in databases, etc.)