CS 227: Homework #1

Assigned: Tuesday, April 5th
Due: Thursday April 14 in class or by 5pm in the CS227 file cabinet in the first floor lobby.

1. *Meeting Scheduling Assistant*. Arranging meetings in an office environment is a tedious process. One common method employed is a series of emails to propose, reject, counter propose, and eventually agree on a time. The effort consumed in such practices motivates the opportunity for forms of automated or semi-automated assistance in scheduling meetings. For example, consider a system that allows users to specify their desiderata (e.g., "sometime next week"), and presents the users with satisfying options for the meeting. Such an assistant will need to have a representation for the Calendar of each participant in a meeting and their preferences. In this exercise, we will start to develop an object-oriented representation of a meeting.
    a. Taking a standard calendar program such as Microsoft Outlook as a reference, specify the classes, subclass relationships amongst them, if any, the slots they should have, and the domain and range constraints for each of the slots. For example, one class that such a representation would need to have is *Appointment.* One of the slots for the class *Appointment* needs to be *Start Time.* The *domain* of the slot *Start Time* is the class *Appointment.* The *range* of the slot *Start Time* is a class called *Time-Point.*
    b. Create a sample instance in this representation by filling out the details of a real or imaginary appointment from your life.
    c. An important piece of knowledge that a meeting scheduler would need to represent and reason with is the knowledge about the participants of a meeting. Such knowledge is usually represented as *constraints.* Later in the course we will be studying the representation of constraints. For the current assignment, simply articulate in English a list of sample constraints. List between three and five constraints. You are free to use examples from your life or your general knowledge about the meeting scheduling preferences of people you know. One example of such a preference is ``I prefer my meetings to be in the afternoon.''

2. *Structured descriptions*. In the chapter on structured descriptions we considered a knowledge representation language that includes concept-forming operators such as FILLS and EXISTS but no role-forming operators. In this question, we extend the language with new concept-forming operators and role-forming operators. Present a formal semantics in the style of Section 9.3.1 of the B&L textbook for the following concept-forming operators:
    a. [SOME r] Role existence. Something with at least 1 r.
    b. [AT-MOST n r] Maximum role cardinality. Something with at most n r's.

    Do the same for the following role-forming operators:
    a. [INVERSE r] Role inverse. So role :Child could be defined as [INVERSE :Parent].
    b. [COMPOSE rl ... rn-1 rn] Role composition. The rn's of the rn-l'S.., of the rl's. So [ALL [COMPOSE :Parent :BrotherlnLaw] Rich] would mean something all of whose uncles are rich (where an uncle is a brother-in-law of a parent).

    Use this semantic specification to show that for any roles r, s, and t, the concept

    [ALL [COMPOSE r s] [SOME t]]

    subsumes the concept

    [ALL r [AND [ALL s [EXISTS 2 t]] [ALL s [AT-MOST 2 t]]]]

    by showing that the extension of the latter concept is always a subset of the extension of the former.

3. *Description Logic*. OWL is a knowledge representation language from the description logic family. The basic concepts introduced in the lectures on frames, structured descriptions, and inheritance reasoning should enable you to read and understand the specification of OWL available at:

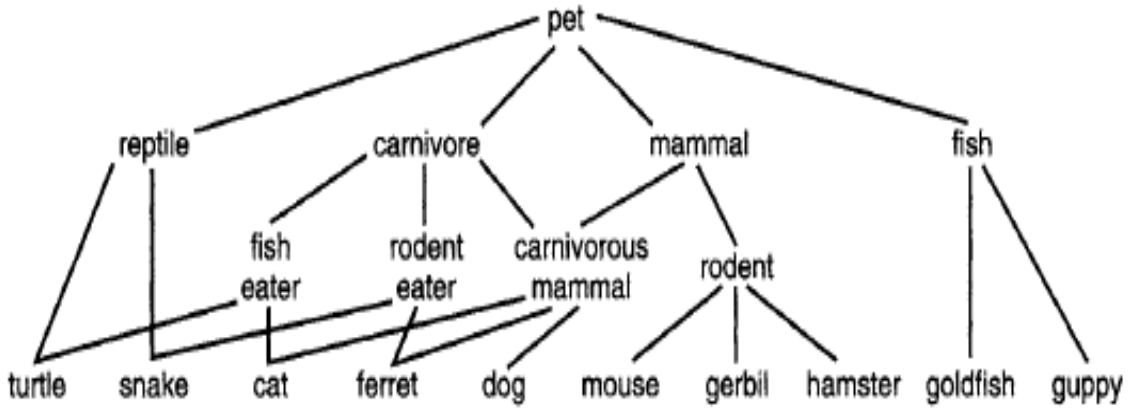
    http://www.w3.org/TR/owl2-primer/

    Read the specification and address the following questions:

a) Represent in OWL using the OWL/XML syntax an individual who is a man with at least three sons who are all unemployed and married to doctors and with at most two daughters who are all professors in physics or math departments. Define any necessary classes that you need to complete these definitions.
b) In section 5 of the OWL specification, the following new constructors are introduced: ObjectIntersectionOf, ObjectUnionOf, ObjectComplementOf, ObjectSomeValuesFrom, ObjectAllValuesFrom, ObjectHasValue, ObjectMaxCardinality, ObjectMinCardinality, ObjectExactCardinality, and ObjectOneOf. Write out the semantics any three of the operators in the style of Section 9.3.1.
c) EquivalentClasses, DisjointClasses and SubObjectPropertyOf are three operators which are used in Section 4 of the OWL specification, but have no equivalent operator in chapter 9 of B&L. Write out the semantics of these operators in the style of Section 9.3.1.

4. *Expressiveness Limitations of Description Logic.*

Many of the disjunctive facts that arise in practice state that a specific individual has one property or another, where the two properties are similar. For example, we may want to represent the fact that a person is either 4 or 5 years old, that a car is either a Chevrolet or a Pontiac, or that a piece of music is either by Mozart or by Haydn. In general, to calculate the entailments of a KB containing such facts, we would need to use a mechanism that considered each case individually, such as Resolution. However, when the conditions being disjoined are sufficiently similar, a better strategy might be to try to sidestep the case analysis by finding a single property that subsumes the disjoined ones. For example, we might treat the original fact as if it merely said that the person is a preschooler, that the car is made by GM, or that the music is by a classical composer, none of which involve explicit disjunctions. Imagine that you have a KB that contains among other things a taxonomy of one-place predicates shown in the Figure below:



Assume that we can use the above taxonomy to find the subsuming cases for disjunctions. Assume that this taxonomy is understood as exhaustive, so that, for example, it implies

Forall x [Mammal(x) → Rodent(x) ∨ CarnivorousMammal(x)].

(a) Given the taxonomy, what single atomic sentence could be used to replace the disjunction (Turtle(fred) ∨ Cat(fred))? Explain why no information is lost in this translation.

(b) What atomic sentence would replace the disjunction (Gerbil(stan) ∨ Hamster(stan))? In this case, information about Stan is lost. Give an example of a sentence that follows from the original KB containing the disjunction, but that no longer follows once the disjunction is eliminated.

(c) What should happen to the disjunction (Dog(sam) ∨ Snake(sam) ∨ Rabbit(sam))?

(d) Present informally a procedure that, given a taxonomy like in the Figure above and a disjunction (P1 (a) ∨ ... ∨ Pn(a)), where the Pi are predicates that may or may not appear in the taxonomy, replaces it by a disjunction containing as few cases as possible.

(e) Argue that a reasoning process that first eliminates disjunctions as we have done in parts (a) through (d) will always be sound.