Solutions (2010)

1. Represent the following statements in the language of description logic and enter the corresponding OWL formulations in Protégé:

(1.1) Every pizza is a meal.

$\mathsf{Pizza} \sqsubseteq \mathsf{Meal}$

(1.2) Pizzas always have two toppings.

$\mathsf{Pizza} \sqsubseteq \geq 2\mathsf{hasTopping}.\top$

(1.3) Every Margherita pizza has a tomato topping.

$\mathsf{Margherita} \sqsubseteq \exists\mathsf{hasTopping}.\mathsf{Tomato}$

(1.4) Everything with a topping is a pizza.

$\exists\mathsf{hasTopping}.\top \sqsubseteq \mathsf{Pizza}$

(1.5) No Margherita pizza has a meat topping.

$\mathsf{Margherita} \sqsubseteq \neg\exists\mathsf{hasTopping}.\mathsf{Meat}$

(1.6) Pizzas, meat and toppings are different things.

$\mathsf{Pizza} \sqsubseteq \neg\mathsf{Meat}$, $\mathsf{Pizza} \sqsubseteq \neg\mathsf{Topping}$, $\mathsf{Topping} \sqsubseteq \neg\mathsf{Meat}$,

(1.7) Property 'has topping is inverse-functional and has domain 'pizza' and range 'topping.'

$\top \sqsubseteq \leq 1\mathsf{hasTopping}^{-}$, $\exists\mathsf{hasTopping}.\top \sqsubseteq \mathsf{Pizza}$, $\top \sqsubseteq \forall\mathsf{hasTopping}.\mathsf{Topping}$

Which of them can be satisfactory represented in RDF(S)? Only (1.1), (1.4) because it can be expressed as 'the domain of hasTopping is Pizza' and the last two statements of (1.7).

2. Sketch a normalised ontology (for use by a library) which covers the following items:

Items to be represented: Books, card catalogue, publications, librarians, photocopies, electronic copies, shelves, rooms, categories of work such as Computer Science, Art History, etc., borrowers, loans;

and provides sufficient concepts and roles to represent the statements:

(2.1) Librarian in charge of publications.

(2.2) Books authored by library staff.

(2.3) Catalogued books.

(2.4) Books on art history of which there are two or more copies in the library.

(2.5) Journals for which there is an electronic copy but no paper copy.

Concepts and properties are organised into the following hierarchy:

self-standing	relations	definables	modifiers
Publication – Book – Journal Catalogue Person PublicationCopy – Photocopy – ElectronicCopy – PaperCopy Location – Shelf – Room Loan	isAuthoredBy hasCategory (func.) hasCopy (inv. func.) isCopyOf (func.) isListedIn isOnLoan (func.) isLoanOf (inv. func.) borrows (inv. func.) isBorrowedBy (func.) canBeFoundAt isLocatedIn (func.) hasStaffRole (func.) isResponsibleFor	Borrower Librarian LibraryStaff	WorkCategory – ComputerScience – ArtHistory – ... StaffRole – LibrarianRole – SecretaryRole – ... StaffResponsibility – Publications – ...

NB: There may be individuals like 'room134', 'Journal of Theoretical Computer Science,' etc. You should put them under the corresponding concepts. For example,

Location
– Shelf
– Room
  – room134 (individual)
Relations:

- isAuthoredBy: domain Book, range Person
  journals contain many articles authored by different persons

- hasCategory: domain Publication, range WorkCategory, functional
  functional since every publication is classified under at most one category

- hasCopy: domain Publication, range PublicationCopy, inverse functional
  inverse functional because a publication copy cannot be a copy of two distinct publications

- isCopyOf: domain PublicationCopy, range Publication, inverse of hasCopy, functional
  functional because hasCopy is inverse functional

- isListedIn: domain Publication, range Catalogue
  library may support several catalogues (e.g., card catalogue, electronic catalogue, etc.); publications can be listed in some of them but we do not assume that every publication is listed in some catalogue;

  our library is equipped with a card catalogue which is referred in what follows to as cardCatalogue;
  in other words, cardCatalogue an instance of the concept Catalogue

- isOnLoan: domain PaperCopy, range Loan, functional
  electronic and photocopies cannot be borrowed; functional since paper copies may be borrowed only once at a time

- isLoanOf: domain Loan, range PaperCopy, inverse of isOnLoan, inverse functional
- borrows: domain Person, range Loan, inverse functional
  inverse functional since its inverse is functional; see below
- isBorrowedBy: domain Loan, range Person, inverse of borrows, functional
  functional since every loan is borrowed by at most one person; see also restrictions below
- canBeFoundAt: domain PaperCopy, range Location
  not functional since several locations of a publication's paper copy may be specified
- isLocatedIn: domain Shelf, range Room, functional
  functional since a shelf cannot be located in two distinct rooms
- hasStaffRole: domain Person, range StaffRole, functional
  helps to untangle hierarchy and make concept trees disjoint
  functional since every library's staff member plays only one role
- isResponsibleFor: domain Person, range StaffResponsibility
  helps to untangle hierarchy and make concept trees disjoint
  not functional since a person may be responsible for many aspects

Restrictions:

- Class(PublicationCopy partial
  restriction(isCopyOf someValuesFrom(owl:Thing)))
  'every publication copy is a copy of something'

  NB. owl:Thing can be replaced (without any change of meaning) with the range of the property (i.e., with Publication)
- Class(Loan partial
  restriction(isLoanOf someValuesFrom(owl:Thing))
  restriction(isBorrowedBy someValuesFrom(owl:Thing)))
  'every loan is a loan of something and is borrowed by somebody'

  NB. in both cases, owl:Thing can be replaced with the range of the property (i.e., with PaperCopy and Person, respectively)

Definables:

- Class(Borrower complete
  Person
  restriction(borrows someValuesFrom(owl:Thing)))
  'a borrower is any person that borrows something'
- Class(Librarian complete
  Person
  restriction(hasStaffRole someValuesFrom(LibrarianRole)))
  'a librarian is any person that has some staff role of a librarian'

- Class(LibraryStaff complete
  Person
  restriction(hasStaffRole someValuesFrom(owl:Thing)))
  'a library staff member is any person that has some staff role'
  NB. we assume that all staff roles are library staff roles

Note that according to the above definitions, Librarian will be re-classified under LibraryStaff (although it is not said explicitly so in the definition).

- Class(LibrarianInChargeOfPublications complete
  Librarian
  restriction(isResponsibleFor someValuesFrom(Publications)))
  'a librarian in charge of publications is
  any librarian that is responsible for publications'

- Class(BookAuthoredByLibraryStaff complete
  Book
  restriction(isAuthoredBy someValuesFrom(LibraryStaff)))
  'a book authored by library staff is
  any book that has at least one library staff member as its author'

- Class(CataloguedBook complete
  Book
  restriction(isListedIn value(cardCatalogue)))
  'a catalogued book is any book that is listed in the library's card catalogue'

- Class(ArtHistoryBookWith2Copies complete
  Book
  restriction(hasCategory someValuesFrom(ArtHistory))
  restriction(hasCopy minCardinality(2)))
  'a book on art history of which there exist two or more copies is
  any book that has some category "Art History" and has at least two copies'

  NB. in the framework of the presented ontology, we cannot specify that those two are book's paper copies because OWL does not support qualified number restrictions

- Class(JournalWithElectronicButNoPaperCopy complete
  Journal
  restriction(hasCopy someValuesFrom(ElectronicCopy))
  complementOf(
  restriction(hasCopy someValuesFrom(PaperCopy))))
  'a journal for which there is an electronic copy but there is no paper copy is any journal that has some electronic copy but does not have any paper copy'

3. Create a DL knowledge base that models the following facts:

(3.1) Mammals are animals.

$\mathsf{Mammal} \sqsubseteq \mathsf{Animal}$

(3.2) Cats are carnivorous mammals.

$\mathsf{Cat} \sqsubseteq \mathsf{Mammal} \sqcap \mathsf{Carnivor}$

(3.3) Elephants are herbivorous mammals.

$\mathsf{Elephant} \sqsubseteq \mathsf{Mammal} \sqcap \mathsf{Herbovore}$

(3.4) Carnivores eat meat.

$\mathsf{Carnivore} \sqsubseteq \exists \mathsf{eat.Meat}$

(3.5) Vertebrate is any animal with a backbone.

$\mathsf{Vertebrate} \sqsubseteq \mathsf{Animal} \sqcap \exists \mathsf{has.Backbone}$

(3.6) Every fish is an animal that lives in water.

$\mathsf{Fish} \sqsubseteq \mathsf{Animal} \sqsubseteq \exists \mathsf{livesIn.Water}$

(3.7) A bird is vertebrate that has wings, legs and lays eggs.

$\mathsf{Bird} \equiv \mathsf{Vertebrate} \sqcap \exists \mathsf{has.Wing} \sqcap \exists \mathsf{has.Leg} \sqcap \exists \mathsf{lays.Egg}$

(3.8) Those who eat meat are carnivores.

$\exists \mathsf{eat.Meat} \sqsubseteq \mathsf{Carnivore}$

(3.9) Rezy is a bird that eats insects and seeds only.

$\mathsf{rezy} : \mathsf{Bird} \sqcap \forall \mathsf{eats.}(\mathsf{Insect} \sqcup \mathsf{Seed})$

4. Using the individuals laura, audrey and donna, concepts Person and NicePerson, and role hasFriend, represent the following knowledge base as an $\mathcal{ALC}$ knowledge base KB:

- Audrey is a person.
- Laura is a nice person.
- Donna is a friend of Laura's.
- A nice person is a person all of whose friends are nice persons.
- Every nice person has a friend.

1. Is the statement "Donna has a friend" a logical consequence of the knowledge base KB? Explain your answer. If the answer is negative, give a model of the KB where the statement is false.
2. Is the statement "Audrey is a friend of Donna's" a logical consequence of the knowledge base KB? Explain your answer. If the answer is negative, give a model of the KB where the statement is false.

The knowledge base KB consists of the following TBox axioms:

- $\mathsf{NicePerson} \quad \equiv \quad \mathsf{Person} \sqcap \forall \mathsf{friend.NicePerson}$
- $\mathsf{NicePerson} \quad \sqsubseteq \quad \exists \mathsf{friend.}\top$

and tree ABox assertions: $\mathsf{audrey} : \mathsf{Person}$, $\mathsf{laura} : \mathsf{NicePerson}$, $(\mathsf{laura}, \mathsf{donna}) : \mathsf{friend}$.

(1) The statement "Donna has a friend" is a logical consequence of the KB, because Donna is a friend of Laura, Laura is a nice person, and so, by the first axiom, Donna must be a nice person. By the second axiom, Donna must have a friend.

(2) The statement "Audrey is a friend of Donna's" is not a logical consequence of the above knowledge base because KB has a model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ such that $(\mathsf{donna}, \mathsf{audrey}) : \mathsf{friend}$ is not satisfied in it:

$$\begin{aligned} \Delta^{\mathcal{I}} &= \{a, l, d\}, \\ \mathsf{Person}^{\mathcal{I}} &= \{a, l, d\}, \\ \mathsf{NicePerson}^{\mathcal{I}} &= \{a, l, d\}, \\ \mathsf{friend}^{\mathcal{I}} &= \{(a, l), (l, d), (d, l)\}, \\ \mathsf{audrey}^{\mathcal{I}} &= a, \\ \mathsf{laura}^{\mathcal{I}} &= l, \\ \mathsf{donna}^{\mathcal{I}} &= d. \end{aligned}$$

5. Consider the following RDF document:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:geo="http://geography.org/#"
         xml:base="http://geography.org/">

    <rdf:Description rdf:about="#UK">
        <rdf:type rdf:resource="#country"/>
    </rdf:Description>

    <rdf:Description rdf:about="#capital_of">
        <rdf:type rdf:resource="&rdf;Property"/>
        <rdf:domain rdf:resource="#city"/>
        <rdf:range rdf:resource="#country"/>
    </rdf:Description>

    <rdfs:Class rdf:about="#country"/>

    <geo:city rdf:about="#london">
        <geo:capital_of rdf:resource="#UK"/>
    </geo:city>

    <rdfs:Class rdf:about="#city"/>
</rdf:RDF>
```

(i) Describe in natural language the content of this document.
(ii) Draw the graph representation of the document.

UK is a country. 'Capital_of' is a property with domain 'city' and range 'country.' 'Country' is a class. London is a city and a (sic!) capital of the UK. City is a class.

6. Consider the following $\mathcal{ALC}$-knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consisting of the TBox $\mathcal{T}$:

$$\begin{aligned} \mathsf{Clownfish} \sqcup \mathsf{Surgeonfish} &\sqsubseteq \mathsf{Fish} \\ \mathsf{Clownfish} \sqcap \mathsf{Surgeonfish} &\sqsubseteq \bot \end{aligned}$$

and the ABox $\mathcal{A}$:

$$\begin{aligned} \mathsf{nemo} &: \mathsf{Clownfish} \sqcap \exists \mathsf{hasColour}.\mathsf{Orange} \\ \mathsf{dory} &: \mathsf{Surgeonfish} \sqcap \forall \mathsf{likes}.\mathsf{Clownfish} \\ \mathsf{darla} &: \mathsf{Person} \\ (\mathsf{dory}, \mathsf{nemo}) &: \mathsf{likes} \\ (\mathsf{dory}, \mathsf{darla}) &: \mathsf{likes} \end{aligned}$$

Is the TBox $\mathcal{T}$ consistent? If so, give an interpretation that is a model of $\mathcal{K}$.

7. Consider the following TBox $\mathcal{T}$:

$$\begin{aligned} \mathsf{Car} &\sqsubseteq \exists \mathsf{producedBy}.\mathsf{CarMaker} \\ \mathsf{CarMaker} &\sqsubseteq \mathsf{Manufacturer} \\ \exists \mathsf{producedBy}.\mathsf{Manufacturer} &\sqsubseteq \mathsf{Product} \end{aligned}$$

Which of the following hold true?

(i) $\mathsf{Car} \sqsubseteq_{\mathcal{T}} \mathsf{Product}$

(ii) $\mathsf{Car} \sqsubseteq_{\mathcal{T}} \mathsf{CarMaker}$

8. Consider the following TBox $\mathcal{T}$:

$$\begin{aligned} A &\sqsubseteq \exists R.B \\ A &\sqsubseteq \forall R.C \end{aligned}$$

Which of the following hold true?

(i) $A \sqsubseteq_{\mathcal{T}} \exists R.(B \sqcap C)$

(ii) $A \sqsubseteq_{\mathcal{T}} \forall R.(B \sqcap C)$

9. Consider the following knowledge base $\mathcal{K}$:

$$\begin{aligned} \mathsf{Human} \sqcap \mathsf{City} &\sqsubseteq \bot \\ \mathsf{Human} &\sqsubseteq \exists \mathsf{livesIn}.\mathsf{City} \\ \mathsf{Human} &\sqsubseteq \mathsf{PhysicalObject} \\ \mathsf{bob} &: \mathsf{Human} \\ \mathsf{john} &: \mathsf{Human} \\ \mathsf{london} &: \mathsf{City} \\ (\mathsf{bob}, \mathsf{london}) &: \mathsf{livesIn} \end{aligned}$$

Is john an instance of the following concepts w.r.t. $\mathcal{K}$?

- Human
- City
- ¬City
- ∃livesIn.City
- ∀livesIn.City