Homework 2

Knowledge Technologies

Exercise 1

“Translate” the following Greek sentences into ALCQO. If you think that the given sentence cannot be translated into ALCQO, then you should give a translation into first-order logic (remember: ALCQO, like all DLs we studied, is a subset of first-order logic).

(α´) Η Ελένη είναι όμορφη.
$\text{Beautiful}(\text{HELEN})$

(β´) Ο Γιάννης είναι όμορφος και πλούσιος.
$\text{Beautiful}(\text{JOHN}) \sqcap \text{Rich}(\text{JOHN})$

(γ´) Ο Πέτρος είναι μυώδης και πλούσιος.
$\text{Muscular}(\text{PETER}) \sqcap \text{Rich}(\text{PETER})$

(δ´) Ο Τίμος είναι μυώδης και ευγενικός.
$\text{Muscular}(\text{TIM}) \sqcap \text{Kind}(\text{TIM})$

(ε´) Σε όλους τους άνδρες αρέσουν οι όμορφες γυναίκες.
$\text{Male} \sqcap \text{likes}.(\text{Beautiful}(\text{Woman}))$

(ς´) ´Ολοι οι πλούσιοι είναι ευτυχισμένοι.
$\text{Rich} \sqsubseteq \text{Happy}$

(ζ´) ´Ολοι οι άνδρες που τους αρέσει μια γυναίκα, στην οποία αρέσουν, είναι ευτυχισμένοι.
$(\forall x)((\exists y)(\text{Man}(x) \sqcap \text{Woman}(y) \sqcap \text{likes}(x, y) \sqcap \text{likes}(y, x) \Rightarrow \text{Happy}(x))$

(η´) ´Ολες οι γυναίκες που τους αρέσει ένας άνδρας, στον οποίο αρέσουν, είναι ευτυχισμένες.
$(\forall x)((\exists y)(\text{Woman}(x) \sqcap \text{Man}(y) \sqcap \text{likes}(x, y) \sqcap \text{likes}(y, x) \Rightarrow \text{Happy}(x))$

(θ΄) Στην Κατερίνα αρέσουν όλοι οι άνδρες, στους οποίους η ίδια αρέσει.
$(\forall x)(\mathrm{likes}(x, \mathrm{KATERINA}) \Rightarrow \mathrm{likes}(\mathrm{KATERINA}, x))$

(ι΄) Στην Ελένη αρέσουν όλοι οι άνδρες που είναι ευγενικοί και πλούσιοι ή μυώδεις και όμορφοι.
$(\forall x)(\mathrm{Man}(x) \sqcap ((\mathrm{Muscular}(x) \sqcap \mathrm{Beautiful}(x)) \sqcup (\mathrm{Kind}(x) \sqcap \mathrm{Beautiful}(x))) \Rightarrow \mathrm{likes}(\mathrm{HELEN}, x))$

(ια΄) Ο Κωστάκης, ο Γιωργάκης και η Ντορούλα είναι μέλη του πολιτικού κόμματος ΔΝΤ.
$\mathrm{PoliticalParty}(\mathrm{DNT}) \sqcap \mathrm{isMemberOf}(\mathrm{KOSTAKIS}, \mathrm{DNT}) \sqcap \mathrm{isMemberOf}(\mathrm{GIORGAKIS}, \mathrm{DNT}) \sqcap \mathrm{isMemberOf}(\mathrm{DOROULA}, \mathrm{DNT})$

(ιβ΄) Κάθε μέλος του κόμματος ΔΝΤ που δεν είναι δεξιός, είναι φιλελεύθερος.
$(\forall x)(\mathrm{PoliticalParty}(\mathrm{DNT}) \sqcap \mathrm{isMemberOf}(x, \mathrm{DNT}) \sqcap \neg\mathrm{RightWinged}(x) \Rightarrow \mathrm{Liberal}(x))$

(ιγ΄) Στους δεξιούς δεν αρέσει ο σοσιαλισμός.
$\forall \mathrm{isRightWinged.Person} \sqsubseteq \neg\mathrm{likes.Socialism}$

(ιδ΄) Σ΄ όποιον δεν αρέσει ο καπιταλισμός, δεν είναι φιλελεύθερος.
$\forall \neg\mathrm{likes}(\mathrm{Person}, \mathrm{Socialism}) \sqsubseteq \neg\mathrm{Liberal}(\mathrm{Person})$

(ιε΄) Στον Κωστάκη δεν αρέσει ό,τι αρέσει στον Γιωργάκη, και του αρέσει ό,τι δεν αρέσει στον Γιωργάκη.
$(\forall x)(\mathrm{likes}(\mathrm{GIORGAKIS}, x) \Rightarrow \neg\mathrm{likes}(\mathrm{KOSTAKIS}, x) \equiv \neg\mathrm{likes}(\mathrm{GIORGAKIS}, x) \Rightarrow \mathrm{likes}(\mathrm{KOSTAKIS}, x))$

(ις΄) Στο Γιωργάκη αρέσει ο σοσιαλισμός και ο καπιταλισμός.
$\mathrm{likes}(\mathrm{GIORGAKIS}, \mathrm{Socialism}) \sqcap \mathrm{likes}(\mathrm{GIORGAKIS}, \mathrm{Capitalism})$

(ιζ΄) Υπάρχει ένα μέλος του ΔΝΤ που είναι φιλελεύθερος αλλά δεν είναι δεξιός.
$(\exists x)(\mathrm{PoliticalParty}(\mathrm{DNT}) \sqcap \mathrm{isMemberOf}(x, \mathrm{DNT}) \sqcap \neg\mathrm{RightWinged}(x) \Rightarrow \mathrm{Liberal}(x))$

(ιη΄) Δίποδο είναι ένα ζώο με ακριβώς δύο πόδια.
$\mathrm{Biped} \equiv \mathrm{Animal} \sqcap (\geq 2\ \mathrm{hasLeg}) \sqcap (\leq 2\ \mathrm{hasLeg})$

(ιθ΄) Τρίγωνο είναι ένα πολύγωνο που έχει ακριβώς τρεις γωνίες και ακριβώς τρεις πλευρές που είναι ευθύγραμμα τμήματα.
$\mathrm{Triangle} \equiv \mathrm{Polygon} \sqcap (\geq 3\ \mathrm{hasAngle}) \sqcap (\leq 3\ \mathrm{hasAngle}) \sqcap$
$(\geq 3\ \mathrm{hasSlide}.(\forall \mathrm{isTypeOf.LineSegment})) \sqcap (\leq 3\ \mathrm{hasSlide}.(\forall \mathrm{isTypeOf.LineSegment}))$

(κ΄) Ορθογώνιο τρίγωνο είναι ένα τρίγωνο που μία από τις γωνίες του είναι ορθή.
OrthogonalTriangle ≡ Triangle ⊓ ∃ hasAngle.(hasDegrees.(isNumber.90))

Exercise 2

Which of the following expressions are syntactically correct in ALCQ and which ones are incorrect?

(a) Person u hasChild
Wrong

(b) ∃hasChild. ≡ Person
Wrong

(c) ∃hasChild.(≥ 1)
Wrong

(d) hasChild ⊑ hasBaby
Wrong

(e) hasChild(ANNA)
Correct

(f) Person ≡ ∃hasChild.⊥ 1
Correct

Exercise 3

Consider the following English sentences:

(a) Wolfgang is a person.
(b) Wolfgang has two distinct pets: Alexis and Yanis (with one “n”).
(c) Alexis and Yanis are animals.
(d) An animal lover is a person which has at least three pets that are animals.
(e) Wolfgang is an animal lover.
(f) Wolfgang is not an animal lover.

Now answer the following questions:

(a) Give an ALCQ knowledge base KB which formalizes the first four of the above sentences and two ALCQ formulas φ and φ’ that formalize the fifth and sixth sentences.

Person(WOLFGANG)
Animal(ALEXIS)
Animal(YANIS)
hasPet(WOLFGANG, ALEXIS)
hasPet(WOLFGANG, YANIS)
AnimalLover ⊑ (Person ⊓ (≥3 hasPet.Animal))
φ = AnimalLover(WOLFGANG)
φ’ = ¬AnimalLover(WOLFGANG)

(b) Can you use tableau techniques to prove that KB |= φ and KB |= φ’ ? For the case or cases where the entailment is not true, prove formally that this is the case.

Tableau technique for φ

1.	Person(WOLFGANG)	given
2.	Animal(ALEXIS)	given
3.	Animal(YANIS)	given
4.	hasPet(WOLFGANG, ALEXIS)	given
5.	hasPet(WOLFGANG, YANIS)	given
6.	AnimalLover ⊑ (Person ⊓ (≥3 hasPet.Animal))	given
7.	AnimalLover(WOLFGANG)	hypothesis
8.	¬ AnimalLover⊔¬(≥ 3 hasPet.Animal)(WOLFGANG)	by ⊑ for 6, 7
9.	¬ Animal lover(WOLFGANG)	by ⊔ for 8: **Clash** by 7

Tableau technique for φ’

1.	Person(WOLFGANG)	given
2.	Animal(ALEXIS)	given
3.	Animal(YANIS)	given
4.	hasPet(WOLFGANG, ALEXIS)	given
5.	hasPet(WOLFGANG, YANIS)	given
6.	AnimalLover ⊑ (Person ⊓ (≥3 hasPet.Animal))	given
7.	¬AnimalLover(WOLFGANG)	hypothesis
8.	AnimalLover⊔¬(≥ 3 hasPet.Animal)(WOLFGANG)	by ⊑ for 6, 7
9.	Animal lover(WOLFGANG)	by ⊔ for 8: **Clash** by 7

Exercise 4

Consider the following English sentences:

(a) Every person is happy if all his children are successful.
(b) All beautiful persons are successful.
(c) Every person is beautiful if one of his/her parents is beautiful, otherwise he/she is ugly.
(d) Aphrodite is a parent of Eros.
(e) Aphrodite is beautiful.
(f) Eros is successful.
(g) Every beautiful person is happy.
(h) Every parent is happy if he/she has no children.

Now answer the following questions:

(a) Write an OWL 2 ontology which encodes sentences (a)-(e) above.

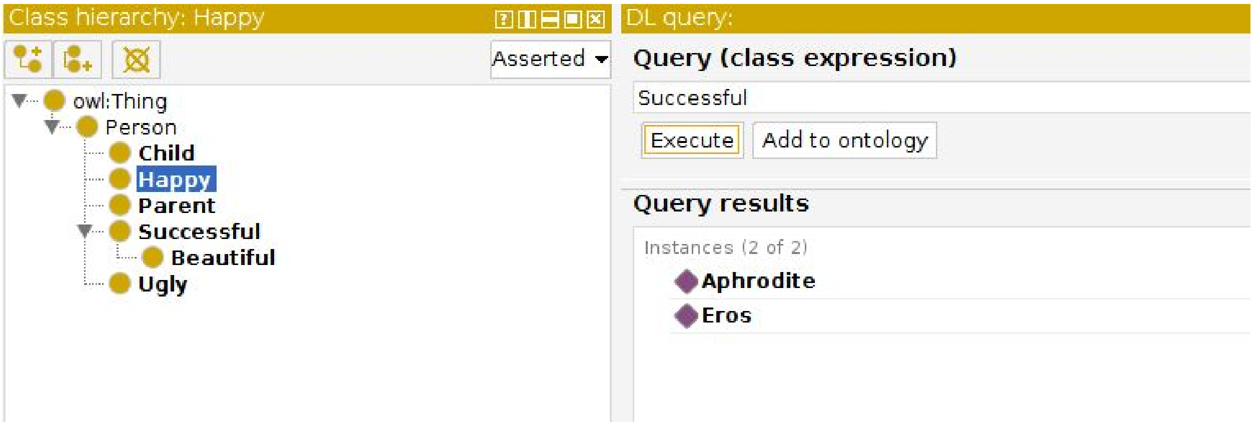
Ontology file created is located inside /ontologies folder called: greekgods.owl

(b) Now formalize the sentences (f)-(h) as OWL 2 axioms.

Which ones of these axioms are entailed by the previous ontology? You do not need to give detailed proofs; only explain why the corresponding entailment relation holds or does not hold and how you can use Protege to show this.

(f) Eros is successful.

- hasParent(APHRODITE, EROS) $\sqcap$ Beautiful(APHRODITE) $\Rightarrow$ Beautiful(EROS)
- Beautiful $\sqsubseteq$ Successful
- Successful(EROS)
  It is entailed. When you query “Successful” you get Eros as one of the results.



```
owl:Thing
  Person
    Child
    Happy
    Parent
    Successful
      Beautiful
    Ugly
```

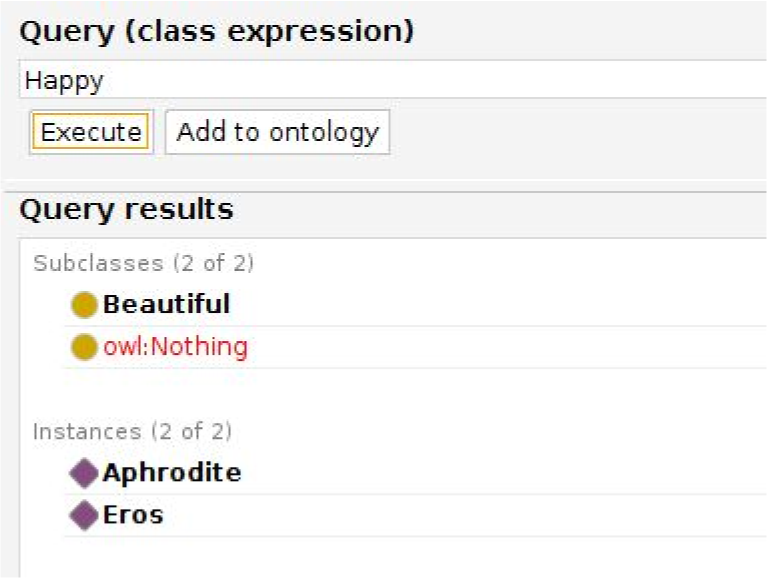
Query (class expression): Successful

Instances (2 of 2)
- Aphrodite
- Eros

(g) Every beautiful person is happy.

A person that is beautiful will have beautiful children (c) and all those children will be successful (b). If all children are successful then the parent is also happy (a), so every beautiful person is happy.

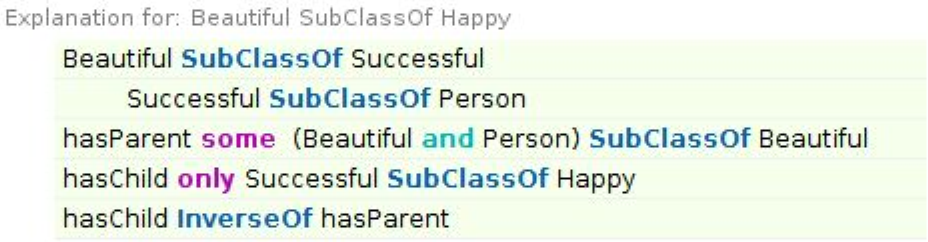
By querying Happy we get Beautiful as subClass result. Protege also provides a detailed explanation.



Query (class expression): Happy

Subclasses (2 of 2)
- Beautiful
- owl:Nothing

Instances (2 of 2)
- Aphrodite
- Eros

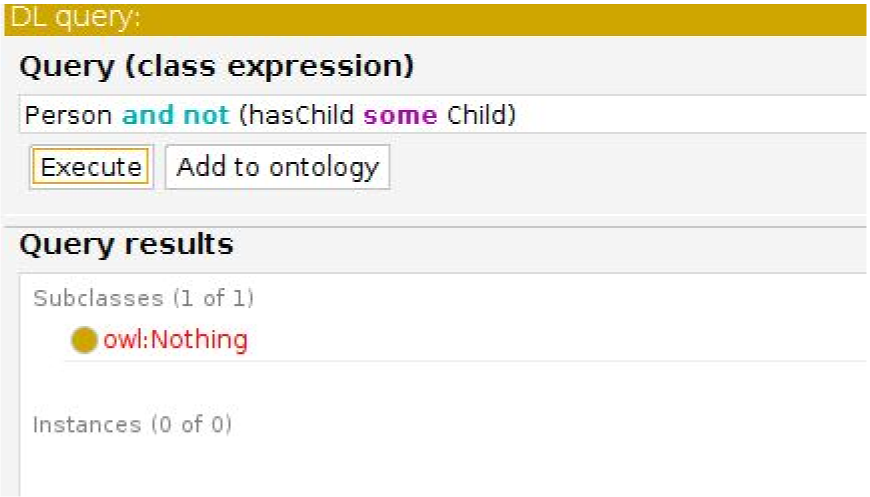


Explanation for: Beautiful SubClassOf Happy

```
Beautiful SubClassOf Successful
        Successful SubClassOf Person
hasParent some  (Beautiful and Person) SubClassOf Beautiful
hasChild only Successful SubClassOf Happy
hasChild InverseOf hasParent
```

(h) Every person is happy if he/she has no children.

It is not entailed.



Query (class expression)

Person and not (hasChild some Child)

Query results

Subclasses (1 of 1)

● owl:Nothing

Instances (0 of 0)

Exercise 5

Consider the following English sentences:

(a) Konstantina, Stella and Roi are members of the club Psiloritis.
(b) Every member of the club Psiloritis who is not a skier is a mountain climber.
(c) Mountain climbers do not like rain.
(d) If someone is a skier then he likes snow.
(e) Konstantina doesn’t like anything that Stella likes.
(f) Stella likes rain and snow.

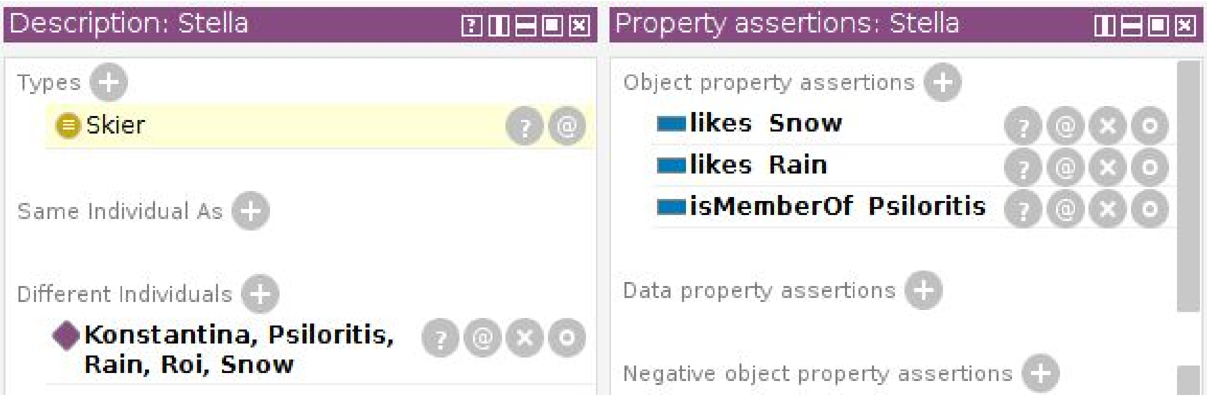
Now answer the following questions:

(a) Give an OWL 2 ontology which formalizes the above sentences.

Ontology file created is located inside /ontologies folder called: psiloritis.owl

(b) Explain what properties and class memberships or non-memberships hold for Konstantina, Stella and Roi as a result of the above sentences and your formalization in OWL 2. Use Protege to verify your claims i.e., discuss what Protege will do with your ontology and how you have verified your claims.

- Stella has property assertions: likes Snow, likes Rain and isMemberOf Psiloritis as a result of (a) and (f). From (d) results that she has the class membership Skier because she likes snow. From (c) results that she has no class membership MountainClimber as MoutainClimbers do not like rain.



Description: Stella

Types
Skier

Same Individual As

Different Individuals
Konstantina, Psiloritis, Rain, Roi, Snow

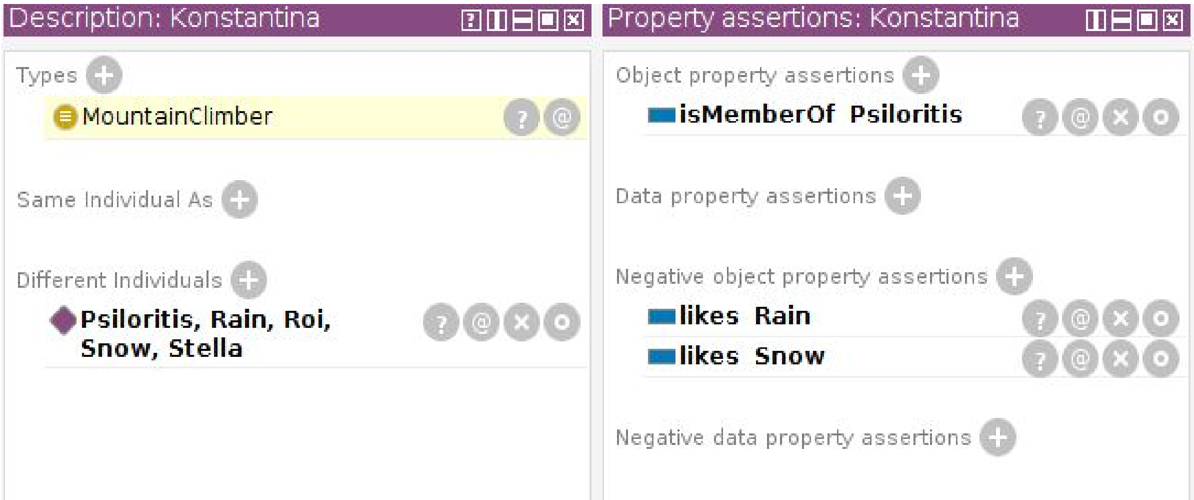
Property assertions: Stella

Object property assertions
likes Snow
likes Rain
isMemberOf Psiloritis

Data property assertions

Negative object property assertions

- Konstantina has property assertions: isMemberOf Psiloritis from (a) and not(likes Rain) and not(likes Snow) from (f) because she dislikes everything that Stella likes.
  This results in her having class membership of MountainClimber as she dislikes rain (c) and dislikes snow (so she's not a skier) (b) .



Description: Konstantina

Types
MountainClimber

Same Individual As

Different Individuals
Psiloritis, Rain, Roi, Snow, Stella

Property assertions: Konstantina

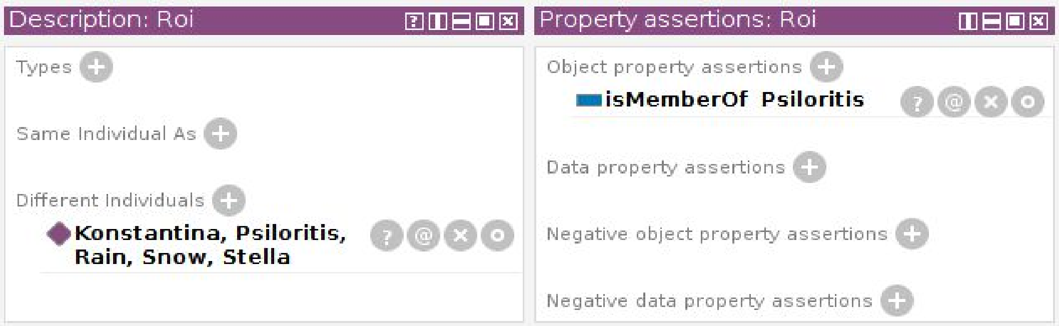
Object property assertions
isMemberOf Psiloritis

Data property assertions

Negative object property assertions
likes Rain
likes Snow

Negative data property assertions

- Roi has property assertions: isMemberOf Psiloritis. We cannot make any more assumptions about Roi as there's no more information whether she likes snow or rain.



Description: Roi

Types

Same Individual As

Different Individuals
Konstantina, Psiloritis, Rain, Snow, Stella

Property assertions: Roi

Object property assertions
isMemberOf Psiloritis

Data property assertions

Negative object property assertions

Negative data property assertions