

1. Koje bi zaključke OWL reasoner mogao da izvuče iz sledećeg RDF segmenta?

```
ex:Pera rdf:type foaf:Person ;  
ex:hasBestFriend <http://www.vets.edu.rs/studenti/Mika> ;  
ex:hasBestFriend <http://www.fon.bg.ac.rs/studenti/MilosMikic> .
```

ukoliko je u ontologiji sa kojom radi property *ex:hasBestFriend* definisan kao **funkcionalni** property (**owl:FunctionalProperty**) čiji su i domen (rdfs:domain) i opseg (rdfs:range) klasa foaf:Person.

Rešenje:

Na osnovu datog RDF segmenta, reasoner bi mogao da zaključi da URIs <http://www.fon.bg.ac.rs/studenti/MilosMikic> i <http://www.vets.edu.rs/studenti/Mika> predstavljaju istu osobu.

Objašnjenje:

Osnovna karakteristika funkcionalnih property-a (owl:FunctionalProperty) je da se najviše jedno pojavljivanje property-a ovog tipa može pridružiti resursima koji čine njegov domen. Na primer, tipični primeri funkcionalnih properties su birthplace i birthdate jer svako ima jedno i samo jedno mesto/datum rođenja.

U ovom primeru je property *ex:hasBestFriend* definisan kao funkcionalni i s obzirom na domen i opseg ovog property-a, njegova interpretacija bi bila da svaka osoba može imati najviše jednog najboljeg prijatelja (osoba ne mora imati najboljeg prijatelja, ali ako ga ima to može biti samo jedna osoba).

2. Koje bi zaključke OWL reasoner mogao da izvuče iz sledećeg RDF segmenta?

```
...
ex:Jim rdf:type foaf:Person ;
      ex:hasAdvisor ex:John .

...
ex:John rdf:type foaf:Person .
...
```

ukoliko je u ontologiji sa kojom radi *ex:hasAdvisor* property definisan kao **inverzni** (**owl:inverseOf**) od *ex:hasPhDStudent* property-a

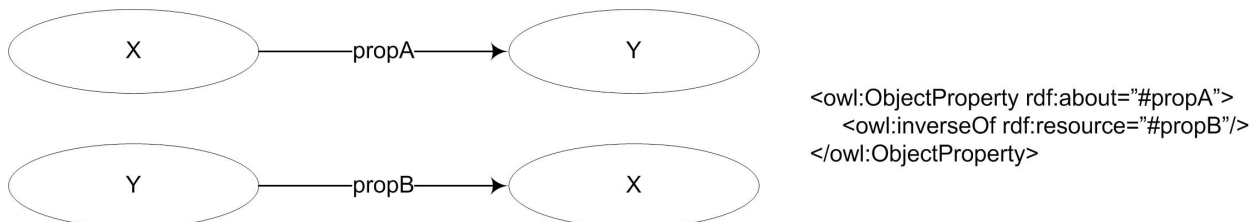
Rešenje:

S obzirom da je property *ex:hasAdvisor* definisan kao inverzni property *ex:hasPhDStudent* property-a, zaključak bi bio sledeći RDF triplet:

```
ex:John rdf:type foaf:Person ;
      ex:hasPhDStudent ex:Jim .
```

Objašnjenje:

Ako su properties *propA* i *propB* definisani kao inverzni, to znači da je domen property-a *propA* predstavlja opseg property-a *propB* i obrnuto (opseg property-a *propA* mora biti domen property-a *propB*). To dalje znači da, subjekat tripleta u kome se property *propA* pojavljuje (u ulozi predikata) mora biti objekat tripleta u kome se property *propB* pojavljuje (kao predikat) i obrnuto (videti sliku; X i Y predstavljaju neke resurse; jednostavnosti radi nisu dati kompletni URIs tih resursa).



3. Koje bi zaključke reasoner mogao da izvuče iz sledećeg RDF segmenta?

```
ex:LoireRegion rdf:type ex:Region ;  
    ex :locatedIn ex:FrenchRegion .
```

...

```
ex:AnjouRegion rdf:type ex:Region ;  
    ex:locatedIn ex:LoireRegion .
```

...

```
ex:Region rdf:type owl:Class .
```

ukoliko je u ontologiji sa kojom radi property *ex:locatedIn* definisan kao **tranzitivni** property (**owl:TransitiveProperty**) čiji su i domen i opseg klasa *ex:Region*.

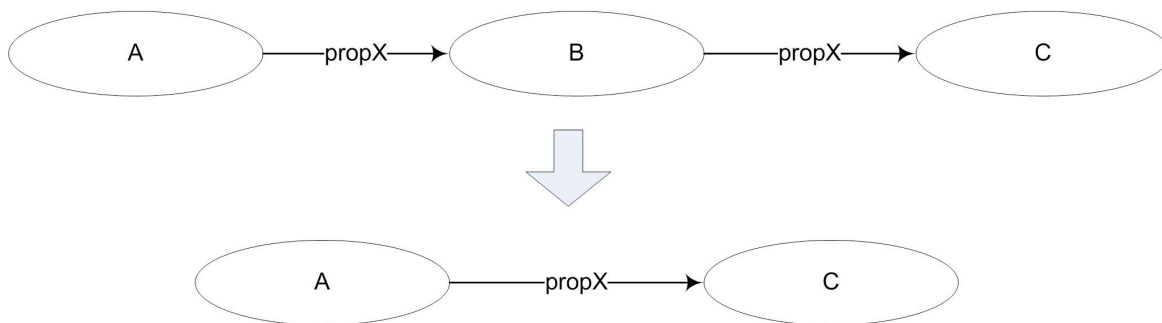
Rešenje:

S obzirom da je property *locatedIn* definisan kao tranzitivni property (*owl:TransitiveProperty*), zaključak bi bio sledeći RDF triplet:

```
ex:AnjouRegion rdf:type ex:Region ;  
    ex:locatedIn ex:FrenchRegion
```

Objašnjenje:

Ovde se radi o klasičnom svojstvu tranzitivnosti, dobro poznatom iz matematike.



4. U ontologiji su definisane klase NonSweetFruit i SweetFruit kao podklase klase Fruit. Koju od navedenih relaciju je potrebno uspostaviti između ovih klasa kako bi se obezbedilo da ni jedna instanca klase SweetFruit ne bude istovremeno i instance klase NonSweetFruit
- owl:disjointWith
 - owl:differentFrom
 - owl:equivalentClass

Rešenje: a)

Objašnjenje: *owl:disjointWith* je primitiva OWL jezika kojom se kaže da neke dve klase (ili u opštem slučaju više klasa) nemaju zajedničke instance.

5. Kreirati mini ontologiju koja će sadržati sledeće elemente:

- a. Klase *Room* i *Furniture*
- b. Klasu *Bedroom* kao podklasu klase *Room*
- c. Klasu *Bed* kao podklasu klase *Furniture*
- d. Property *furnishedWith* čiji je domen klasa *Room* a opseg klasa *Furniture*
- e. Ograničenje na property *furnishedWith* u kontekstu klase *Bedroom*, tako da bar jedna vrednost ovog property-a bude instanca klase *Bed*

Navesti naziv ograničenja korišćenog u tački (e) i objasniti njegovo značenje.

Rešenje:

Korišeno ograničenje je *owl:someValuesFrom*. Ovo ograničenje se naziva *egzistencijalno ograničenje* jer ono zahteva da u kontekstu klase na koju se primenjuje (u datom zadatku klasa *Bedroom*) postoji bar jedno pojavljivanje property-a na koje se odnosi (u ovom zadatku *furnishedWith* property) čija je vrednost instanca zadate klase (u ovom slučaju klase *Bed*).

Ovim se ne sprečava da instance klase na koju se ovo ograničenje primenjuje imaju veći broj property-a datog tipa sa nekim drugim vrednostima: u ovom zadatku, da instance klase *Bedroom* imaju više pridruženih *furnishedWith* property-a čija vrednost nije instanca klase *Bed* već klase *Furniture* ili neke druge njene podklase.

6. Kreirati mini ontologiju koja će sadržati sledeće elemente:
- Klasu *Dish*, *VegeterianDish*, *Ingredient* i *VegeterianIngredient*
 - Klasu *VegeterianDish* definisati kao podklasu klase *Dish*, a *VegeterianIngredient* kao podklasu klase *Ingredient*.
 - Property *hasIngredient* čiji je domen klasa *Dish* a opseg klasa *Ingredient*.
 - Ograničenje na property *hasIngredient* u kontekstu klase *VegeterianDish*, tako da vrednosti ovog property-a (u kontekstu ove klase) mogu biti samo instance klase *VegeterianIngredient*.
- Navesti naziv ograničenja korišćenog u tački (c) i objasniti njegovo značenje.

Primer primene: preporuka restorana na osnovu specifičnih preferenci u ishrani; npr., osobi koja je vegeterijanac predložiti samo restorane sa širokom ponudom vegeterijanskih jela (*VegeterianDish*)

Rešenje:

Korišćeno ograničenje je *owl:allValuesFrom*. Ovo ograničenje se naziva *univerzalno ograničenje*. Ono zahteva da u kontekstu klase na koju se primenjuje (u datom zadatku klasa *VegeterianDish*) sva pojavljivanja property-a na koje se odnosi (u ovom zadatku *hasIngredient* property) sadrže kao svoju vrednost isključivo instance zadate klase (u ovom slučaju klase *VegeterianIngredient*).

Ovim se ne sprečava da instance klase na koju se ovo ograničenje primenjuje nemaju pridružen property datog tipa (na primer, u ovom zadatku, da nekoj instanci klase *VegeterianDish* nije pridružen nijedan *hasIngredient* property).

Sličan primer primene:

Preporuka restorana ili specifičnog jela/namirnice na osnovu raspoloživih podataka o alergijama na pojedine vrste hrane koje osoba može imati. Na primer, nekome ko je alergičan na gluten, predložiti restorane/prodavnice koje nude gluten-free jela/namirnice.

Primer definisanja skupa (tj. *owl:Class*) gluten-free namirnica (*GlutenFreeFoodItem*) kao komplementa (***owl:complementOf***) skupa svih namirnica (*FoodItem*) i onih koje sadrže gluten (*FoodItemWithGluten*)

