i Front Page

UNIVERSITY OF OSLO

Faculty of Mathematics and Natural Sciences Examination in INF3580/INF4580

Day of examination: 15. June 2017 Examination hours: 9:00–13:00 Place: Sal D Silurveien 2

Permitted aids: Any printed or written course material

The exam consists of 5 sections with equal weight.

i Section 1: RDF and R2RML

Section 1: RDF and R2RML

1 RDF, triples and prefixes

Here are some triples with long, hard to read URIs. Write the same triples in Turtle notation, introducing some prefixes. (You can use standard prefixes, or invent your own.)

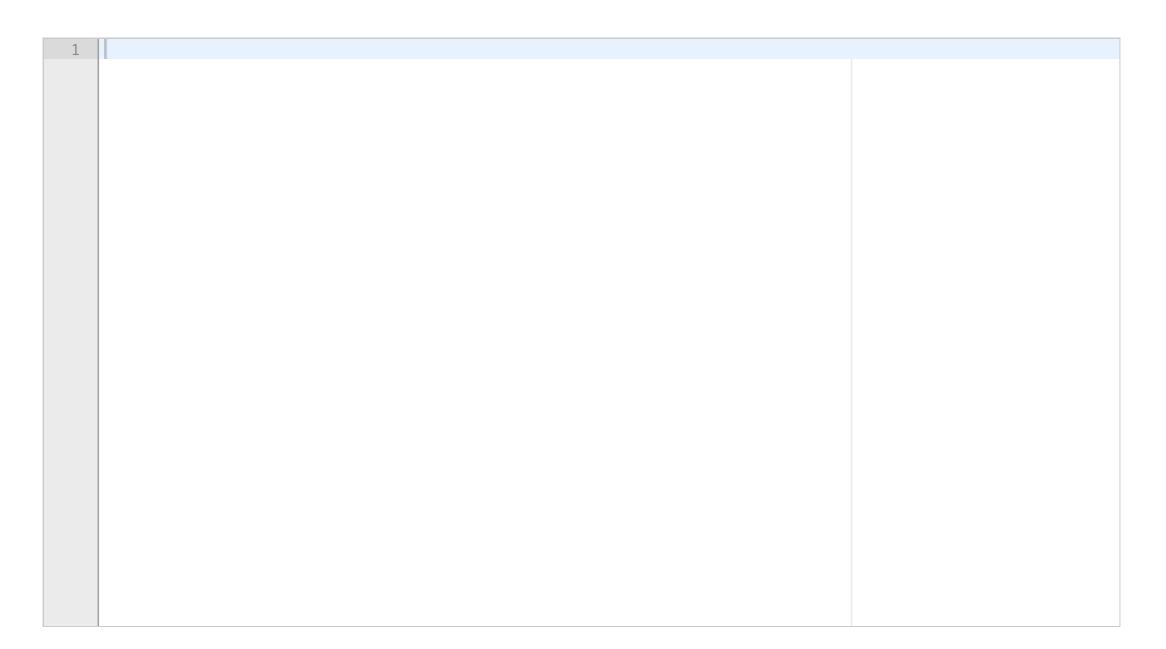
http://dbpedia.org/ontology/TelevisionShow http://dbpedia.org/ontology/Work .

http://dbpedia.org/resource/How_I_Met_Your_Mother http://dbpedia.org/ontology/TelevisionShow.

 "How I Met Your Mother" . .

http://dbpedia.org/ontology/starring http://dbpedia.org/ontology/starring <a href="

<a href="http://dbpedia.org/resource/How_I_Met_Your_Mother" http://xmlns.com/foaf/name "How I Met Your Mother"^^http://www.w3.org/2001/XMLSchema#string.



Maximum marks: 4

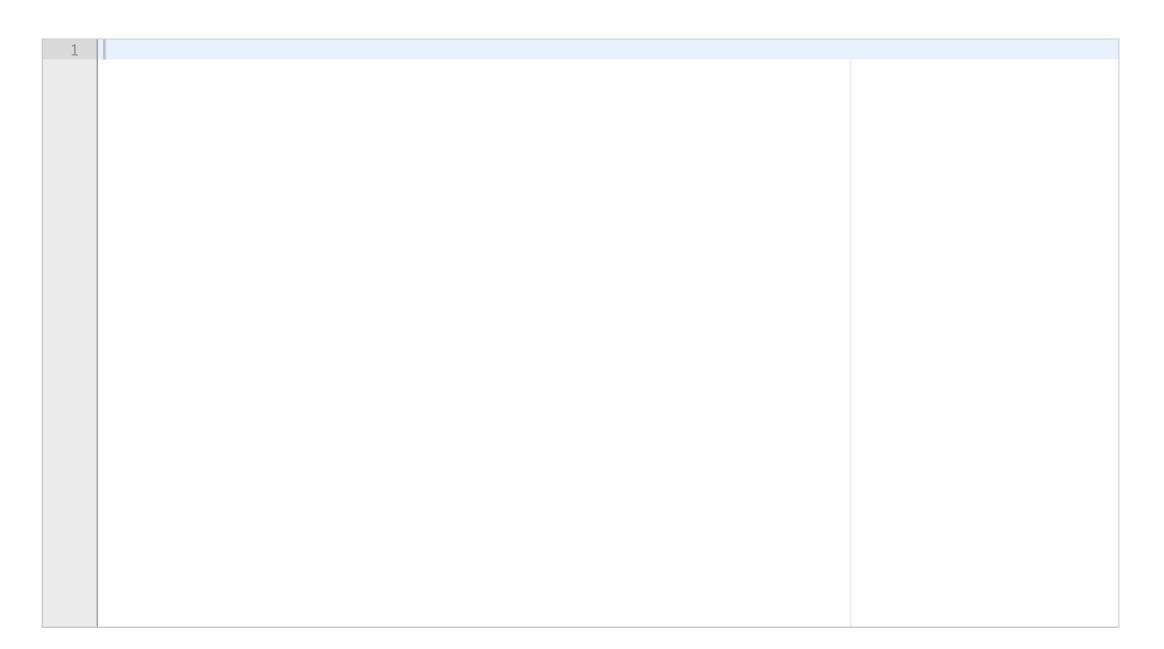
2 Triples for R2RML mapping

Given the following table in a relational database:

Table **SERIES**

Title	Genre	Seasons	Episodes
How_I_Met_You	RMotaetic_Come	59 y	208
Modern_Family	Sitcom	7	159

```
And the following mapping definition (we assume missing prefixes have been already defined):
<TriplesMap1> a rr:TriplesMap;
 rr:logicalTable [rr:SQLQuery "Select * from SERIES"];
 rr:subjectMap
    rr:template "http://dbpedia.org/resource/{Title}";
    rr:class <a href="http://dbpedia.org/ontology/TelevisionShow">http://dbpedia.org/ontology/TelevisionShow</a>;
 ];
 rr:predicateObjectMap
    rr:predicate <a href="http://dbpedia.org/ontology/numberOfEpisodes">http://dbpedia.org/ontology/numberOfEpisodes</a>;
    rr:objectMap [rr:column "Episodes"; rr:datatype xsd:integer;]
 rr:predicateObjectMap
    rr:predicate <a href="http://dbpedia.org/ontology/numberOfSeasons">http://dbpedia.org/ontology/numberOfSeasons</a>;
    rr:objectMap [rr:column "Seasons"; rr:datatype xsd:integer;]
 rr:predicateObjectMap
    rr:predicate <http://dbpedia.org/ontology/genre>;
    rr:objectMap [rr:template "http://dbpedia.org/resource/{Genre}"]
Write down the triples generated by the mapping applied to the table "SERIES".
```



Maximum marks: 6

R2RML mapping from triples

Give the following database table:

Table **PLAYED-ROLE**

Actor	Role	
Alyson_Hannigan	Lily_Aldrin	
Joshua_Radnor	Ted_Mosby	

Define an R2RML mapping that generates the following triples:

- http://dbpedia.org/resource/Alyson_Hannigan http://dbpedia.org/ontology/playRole
- http://dbpedia.org/resource/Lily_Aldrin.
- http://dbpedia.org/ontology/playRole
- http://dbpedia.org/resource/Ted_Mosby.

1	

Maximum marks: 6

4 Blank nodes

In this question, wrong answers will give 0 points.

Which of these are true and which false?

	False	True
Blank nodes can be used in the predicate position		
Blank nodes can be used only in the subject position		0
Blank nodes have URIs		0
Blank nodes represent resources		0
Blank nodes do not have a global identifier	0	

Maximum marks: 4

i Section 2: SPARQL

Section 2: SPARQL

Assume we have the following prefixes:

dbo - http://dbpedia.org/ontology/

dbr - http://dbpedia.org/resource/

xsd - http://www.w3.org/2001/XMLSchema

foaf - http://xmlns.com/foaf/

and the following RDFS classes and properties based on dbpedia:

dbo:Actor - The class of all actors

dbo:Person - The class of all people

dbo:TelevisionShow - The class of all TV shows

dbo:TelevisionEpisode - The class of all TV episodes

dbo:showName - Relationship between a TV show and a string denoting the TV show name

dbo:title - Relationship between a TV episode and a string denoting the episode's title

foaf:name - Relationship between a person and a string denoting the person name

dbo:starring - Relationship between a TV show or film and an actor

dbo:guest - Relationship between a TV episode and a person

dbo:releaseDate - Relationship between a TV show or TV episode and a date (xsd:date) representing the date the TV show/episode was first aired

dbo:completionDate - Relationship between a TV show and a date (xsd:date) representing the date the TV show was last aired

dbo:numSeasons - Relationship between a TV show and a integer literal denoting the number of seasons of the TV show

dbo:numEpisodes - Relationship between a TV show and a integer literal denoting the number of episodes of the TV show

Below are some example triples:

dbr:How_I_Met_Your_Mother	а	dbo:TelevisionShow;
	dbo:showName	"How I Met Your Mother"^^xsd:string;
	dbo:releaseDate	"2005-09-19"^^xsd:date;
	dbo:starring	dbr:Josh_Radnor ;
	dbo:numSeasons	9^^xsd:integer;
	dbo:numEpisodes	208^^xsd:integer.

dbr:Josh_Radnor	а	dbo:Actor;
	foaf:name	Josh Radnor .

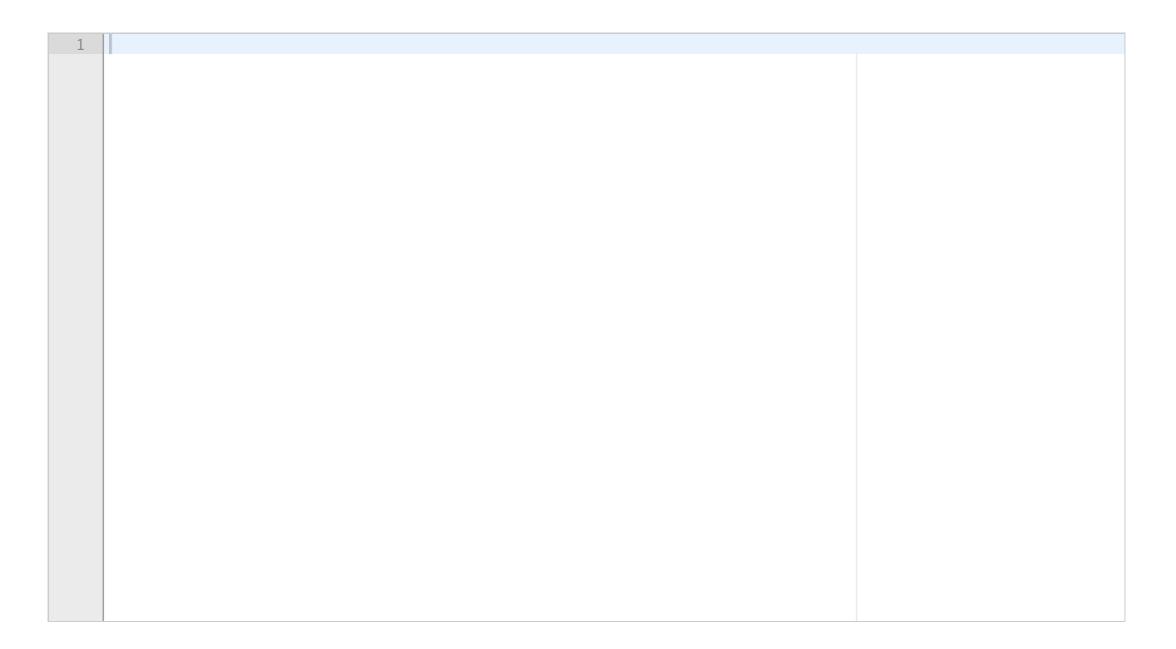
dbr:The_Stamp_Tramp	а	dbo:TelevisionEpisode;	
	dbo:series	dbr:How_I_Met_Your_Mother;	
	dbo:guest	dbr:Joe_Manganiello .	

dbr:Joe_Manganiello	а	dbo:Actor;
	foaf:name	Joe Manganiello .

TV Shows in 2016

5

Write a SPARQL query that lists the names of TV shows with episodes during 2016.

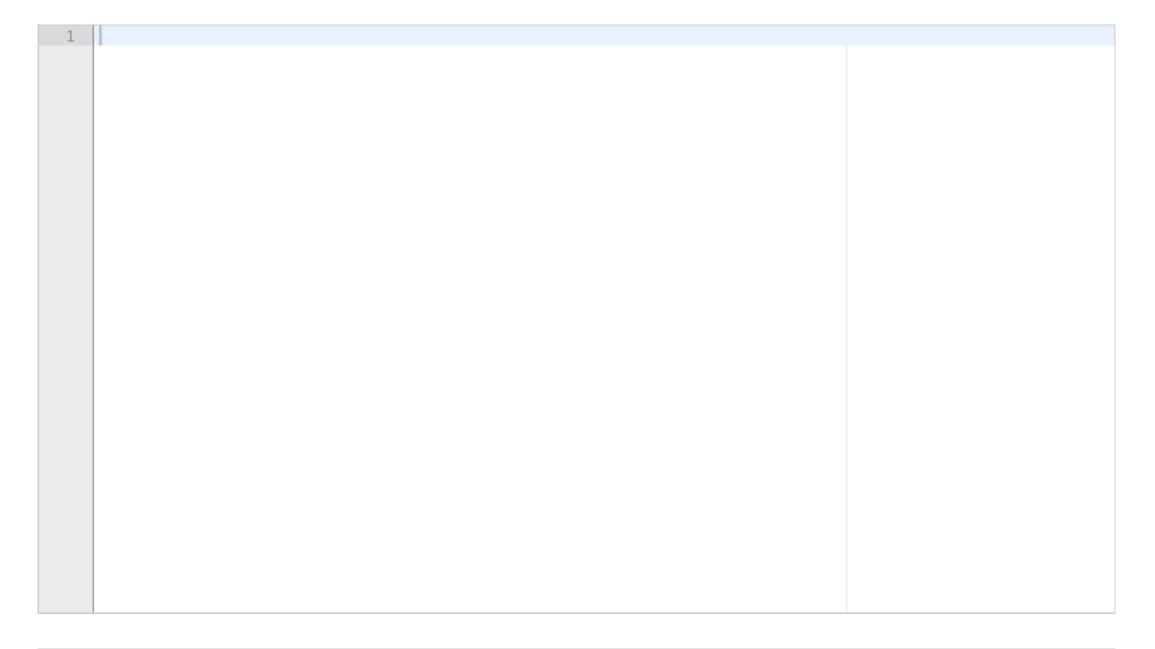


Maximum marks: 4

6 Shows with many guests

Write a SPARQL query that lists the TV episodes and number of guests, for the episodes with more than 10 guests.

Fill in your answer here

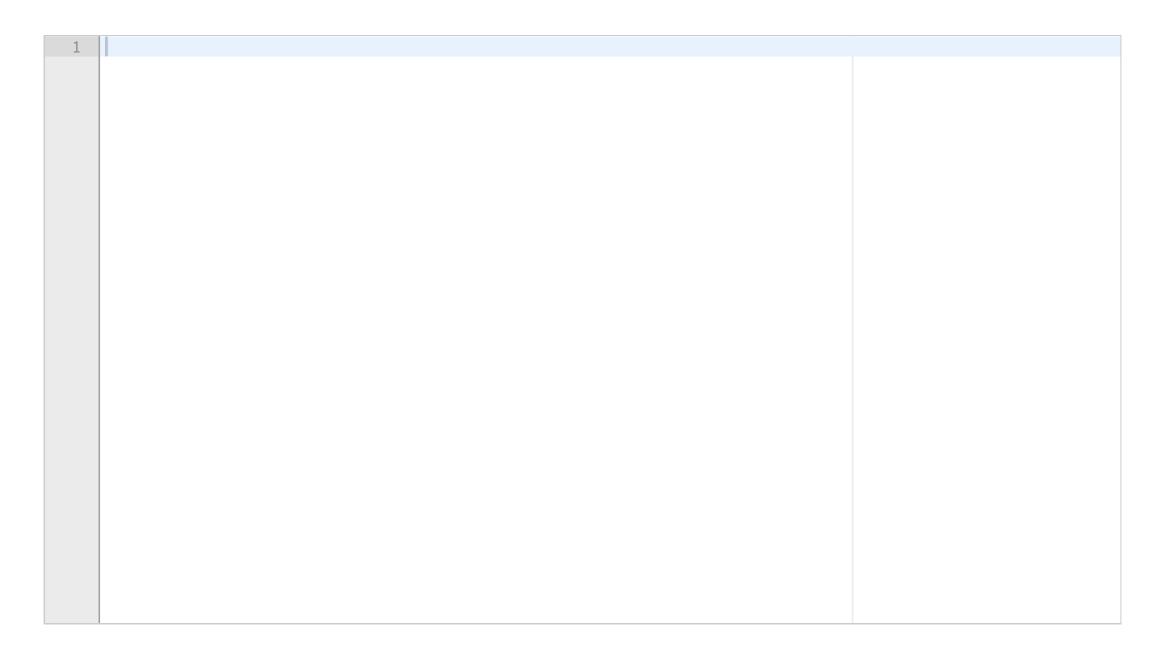


Maximum marks: 4

7 Ongoing TV Shows

Write a SPARQL query that constructs an RDF-graph with triples of the form < x rdf:type dbo:OngoingTVShow> for every TV show "x" that does not have a completion date.

Fill in your answer here

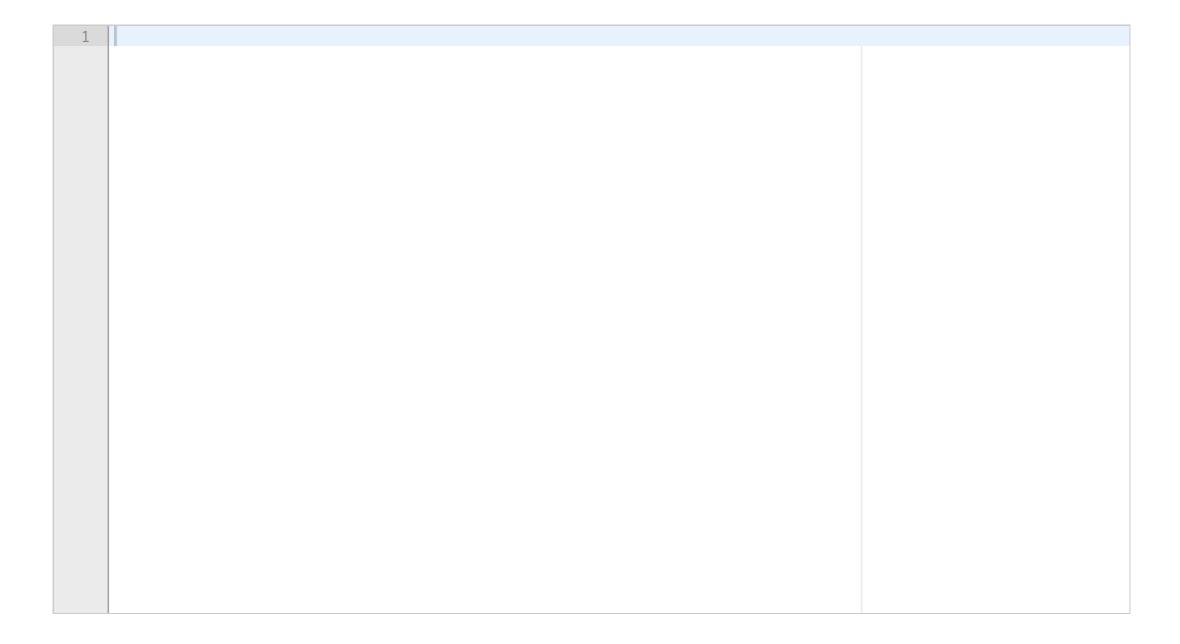


Maximum marks: 4

Long running shows

Write a SPARQL query that lists the TV shows that either have more than 200 episodes or were on air for more than 15 years. You can use the function year(...) in a FILTER to get the year from a xsd:date.

Fill in your answer here



Maximum marks: 4

9 SPARQL Entailment regimes

Consider the following subgraph from dbpedia:

dbr:Josh Radnor owl:sameAs freebase:Josh Radnor.

dbo:Actor rdfs:subClassOf dbo:Person.

dbo:starring rdfs:range dbo:Actor.

dbo:guest rdfs:range dbo:Person.

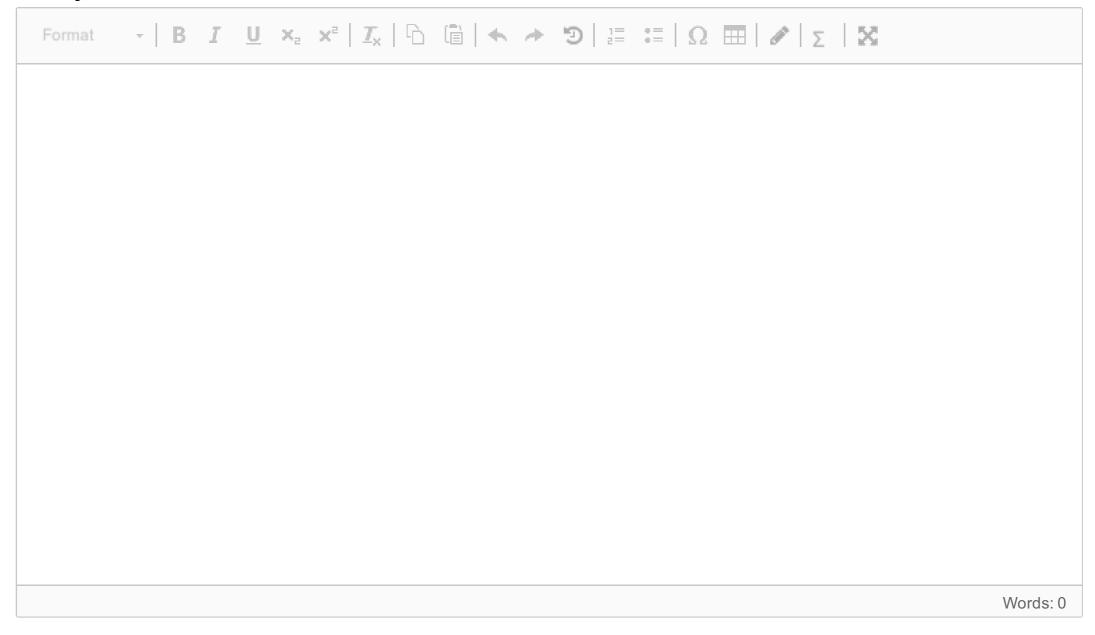
dbr:How_I_Met_Your_Mother dbo:starring dbr:Alyson_Hannigan .

dbr:How_I_Met_Your_Mother dbo:guest dbr:Joe_Manganiello.

Write the result of running the following SPARQL query over the above graph using: (a) Simple entailment, (b) RDF entailment, (c) RDF Schema entailment, and (d) OWL 2 Direct Semantics entailment.

SELECT ?person WHERE { ?person rdf:type dbo:Person }

Fill in your answer here



Maximum marks: 4

i Section 3: RDFS inference

Section 3: RDFS inference

Assume we have the following prefixes

@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns# .

@prefix rdfs: http://www.w3.org/2000/01/rdf-schema#.

@prefix : <http://www.ifi.uio.no/banking#> .

and let Γ be the following set of triples:

(7)

(1) :Account rdfs:subClassOf :BankService. :SavingsAccount rdfs:subClassOf :Account . (2) :DebitAccount rdfs:subClassOf :Account. (3) (4) :hasAccount rdfs:domain :Customer. (5) rdfs:range :hasAccount :Account. rdfs:subPropertyOfnasService. (6) :hasAccount

:hasSavingsAccountfs:subPropertyOfnasAccount.

INF3580/4580 v17

(8) :hasDebitAccountrdfs:subPropertyOfnasAccount .

(9)	:sa	rdf:type	:SavingsAccount
(10)	:sandra	:hasSavingsAc	coust.
(11)	:peter	:hasAccount	:ba .
(12)	_:x	:hasService	:service .

You can use the numbers in the left-most column to reference the triples in your derivations in the following exercises.

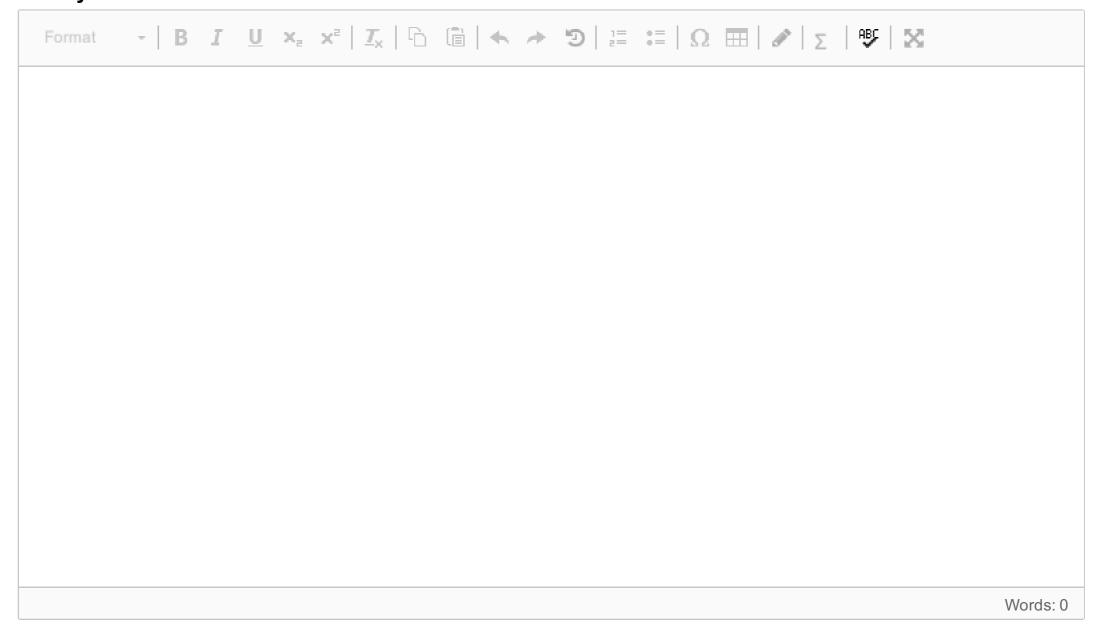
10 Type of :sa

For the triple

:sa rdf:type :BankService .

either give a derivation from Γ using the rules of RDFS and simple entailment or explain why this is not possible if no such derivation exists. Also indicate whether the triple is entailed by Γ or not.

Fill in your answer here



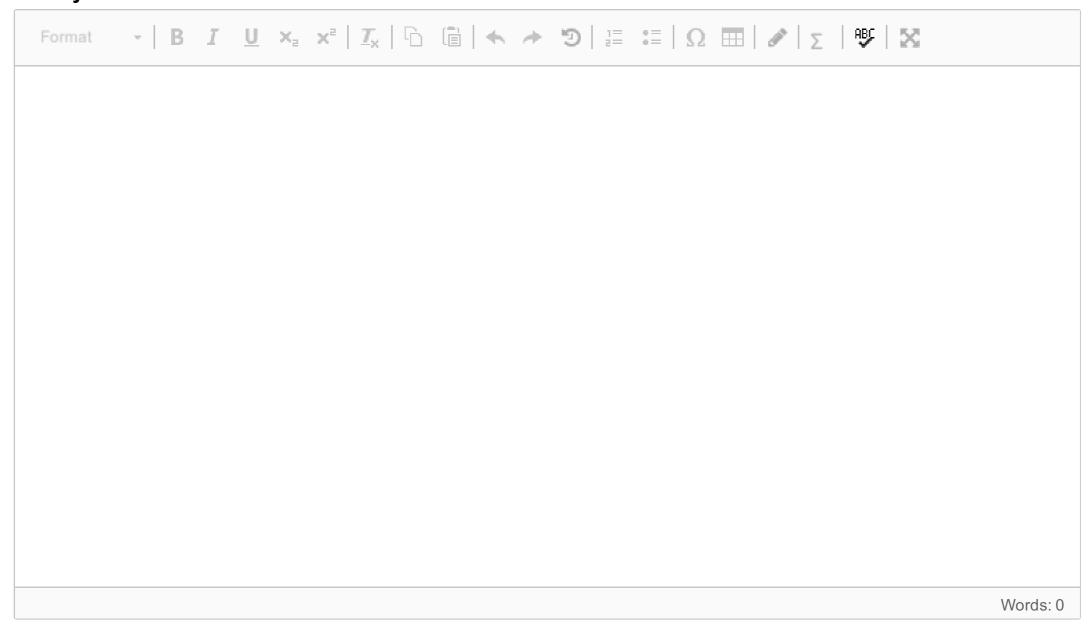
Maximum marks: 3

11 Type of :peter

For the triple

:peter rdf:type :Customer .

either give a derivation from Γ using the rules of RDFS and simple entailment or explain why this is not possible if no such derivation exists. Also indicate whether the triple is entailed by Γ or not.



Maximum marks: 3

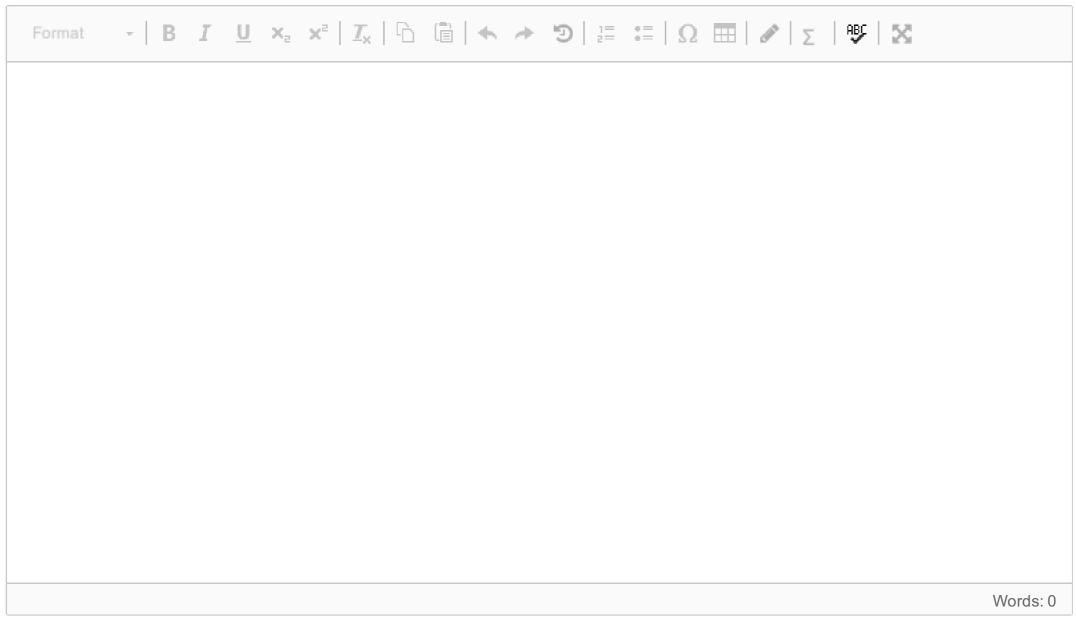
12 _:y has service

For the triple

_:y :hasService _:y .

either give a derivation from Γ using the rules of RDFS and simple entailment or explain why this is not possible if no such derivation exists. Also indicate whether the triple is entailed by Γ or not.

Fill in your answer here

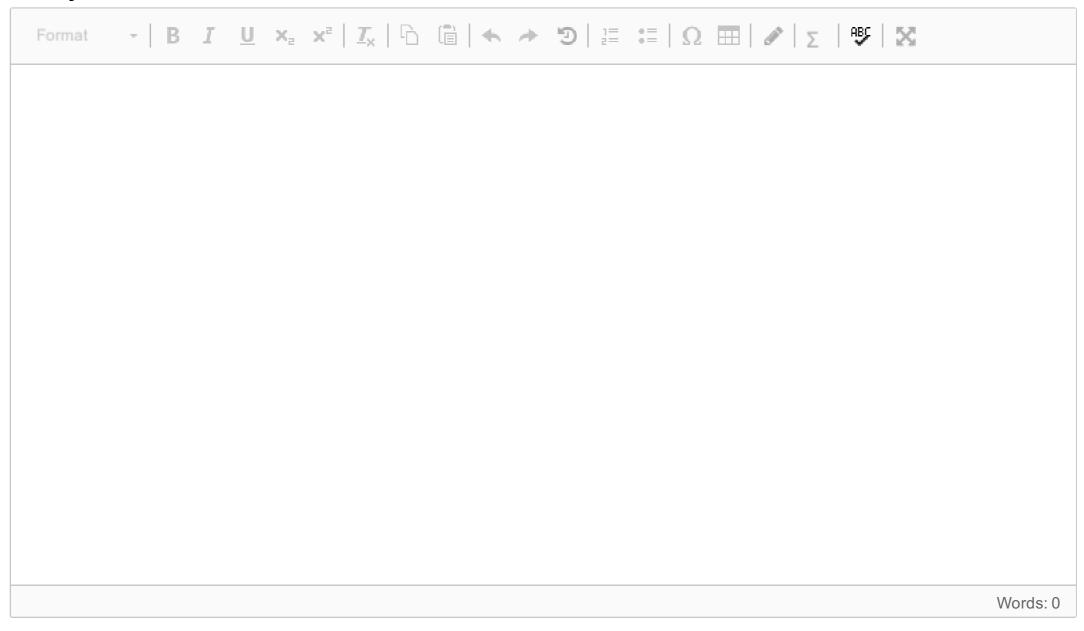


Maximum marks: 4

13 Entailed but not derivable

Write down a triple that is entailed by Γ but which is impossible to derive using the rules of RDFS and simple entailment and explain why it is entailed but not derivable.

Fill in your answer here



Maximum marks: 5

14 :sandra has account

Assume we have the same prefixes as before, and let Γ' be the set of triples

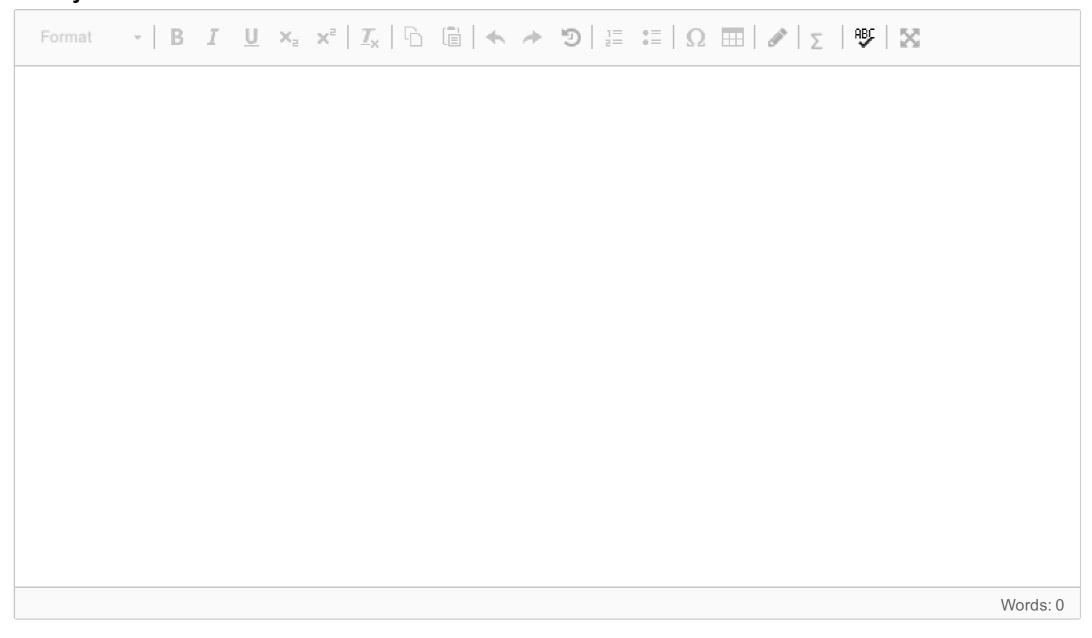
(1) :hasPrivateLoan rdfs:subPropertyOf _:r .(2) :hasCorporateLoan rdfs:subPropertyOf _:r .

(3) _:r rdfs:subPropertyOf :hasService .(4) :sandra :hasPrivateLoan :loan .

For the triple

:sandra :hasService :loan .

either give a derivation from Γ' using the rules of RDFS and simple entailment or explain why this is not possible if no such derivation exists.



Maximum marks: 5

Section 4: Description Logic and OWLSection 4: Description Logic and OWL

In the following exercises you can use Manchester syntax to write DL-axioms. The conversion between DL-syntax and Manchester syntax is given in the following table:

Constructor	DL-syntax	Manchester syntax
Bottom		Nothing
Top	T	Thing
Intersection	$C \sqcap D$	C and D
Union	$C \sqcup D$	C or D
Complement	$\neg C$	$\operatorname{not} C$
Closed class	$\{a,b,c\}$	$\{a,b,c\}$
Existential restriction	$\exists R.C$	R some C
Universal restriction	$\forall R.C$	$R ext{ only } C$
Min. cardinality restriction	$\geq n R.C$	$R \min n C$
Max. cardinality restriction	$\leq n R.C$	$R \max n C$
Eq. cardinality restriction	= nR.C	R exactly nC
Self restriction	$\exists R. \text{Self}$	R Self
Property chain	$R \circ P$	RoP
Inverse property	R^-	inverse R
Property complement	$\neg R$	$\operatorname{not} R$
Axiom	DL-syntax	Manchester syntax
Subclass of	$C \sqsubseteq D$	C SubClassOf D
Equivalent class	$C \equiv D$	C Equivalent To D
Subproperty of	$R \sqsubseteq P$	C SubPropertyOf D
Equivalent property	$R \equiv P$	R Equivalent To P

i Classes, properties and individuals

In the following exercises, use this vocabulary:

Classes:

- Computer
- SuperComputer
- CPU
- Core
- RAM
- Motherboard

Properties:

- hasPart
- hasCore
- isConnectedTo

Individuals:

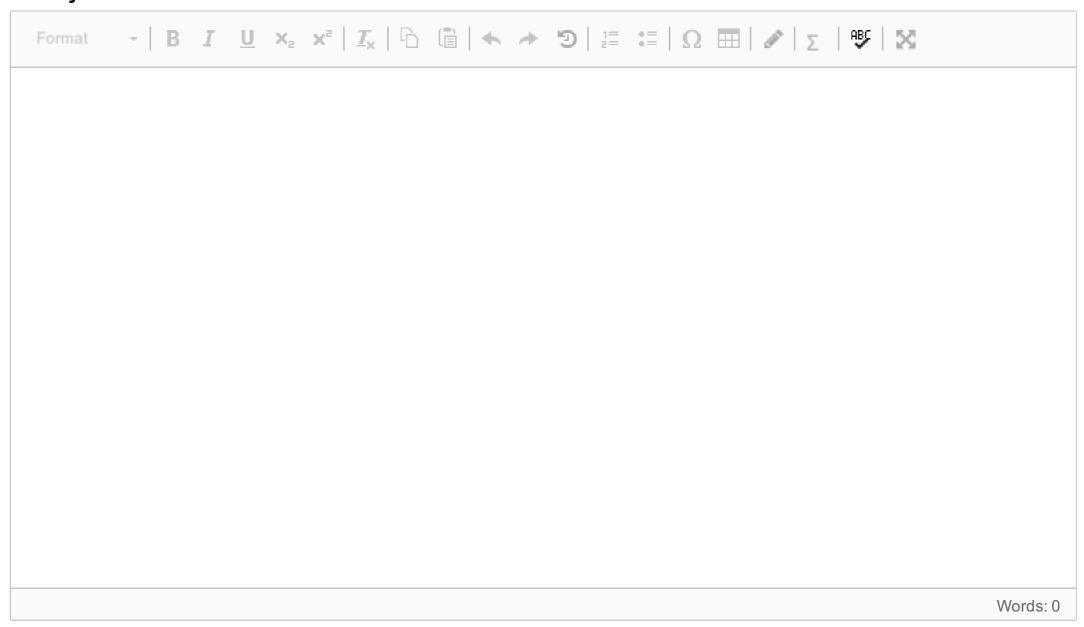
• superCPU

A computer can e.g. have a CPU as part, and a CPU can have cores. A CPU can e.g. be connected to a motherboard. But CPUs are not parts of motherboards, they are connected to motherboards.

15 Computer, CPU

Write a DL-axiom stating that a computer is precisely something that has a CPU.

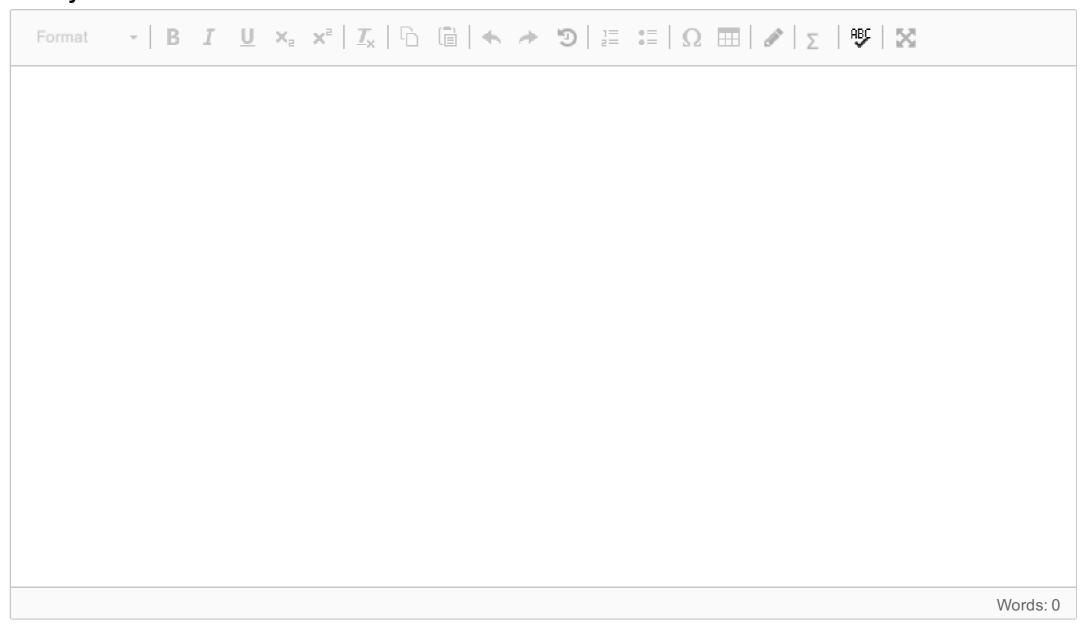
Fill in your answer here



Maximum marks: 2

16 Parts of computer

Write a DL-axiom stating that all parts of a computer are either motherboards or things connected to motherboards.

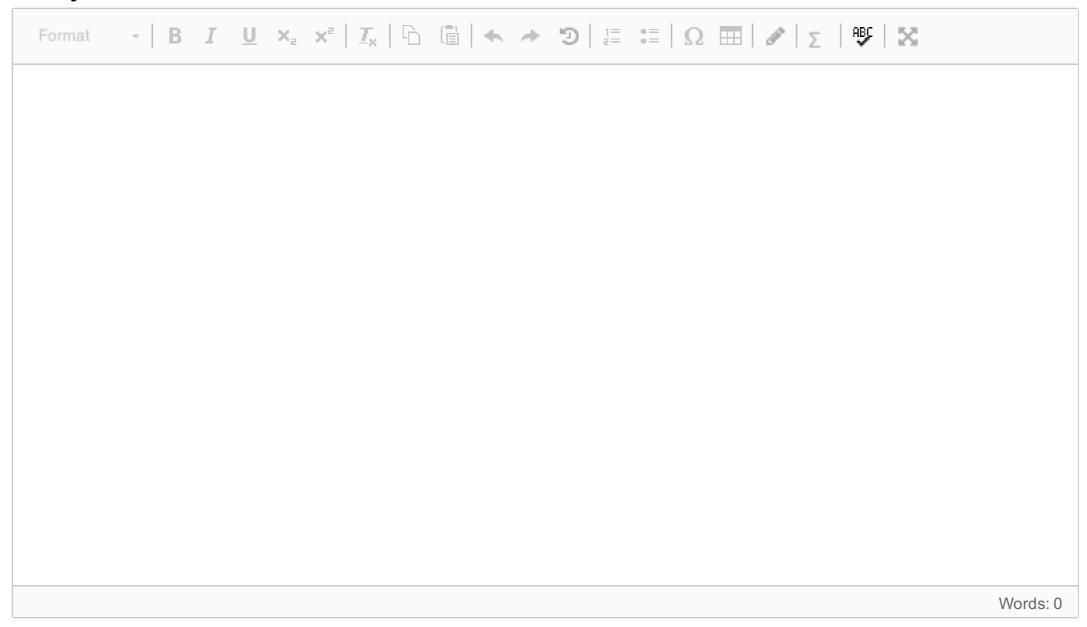


Maximum marks: 4

17 Super-computer

Write a DL-axiom stating that if a computer either has more than 2 CPUs or a CPU with more than 8 cores, then it is a super-computer.

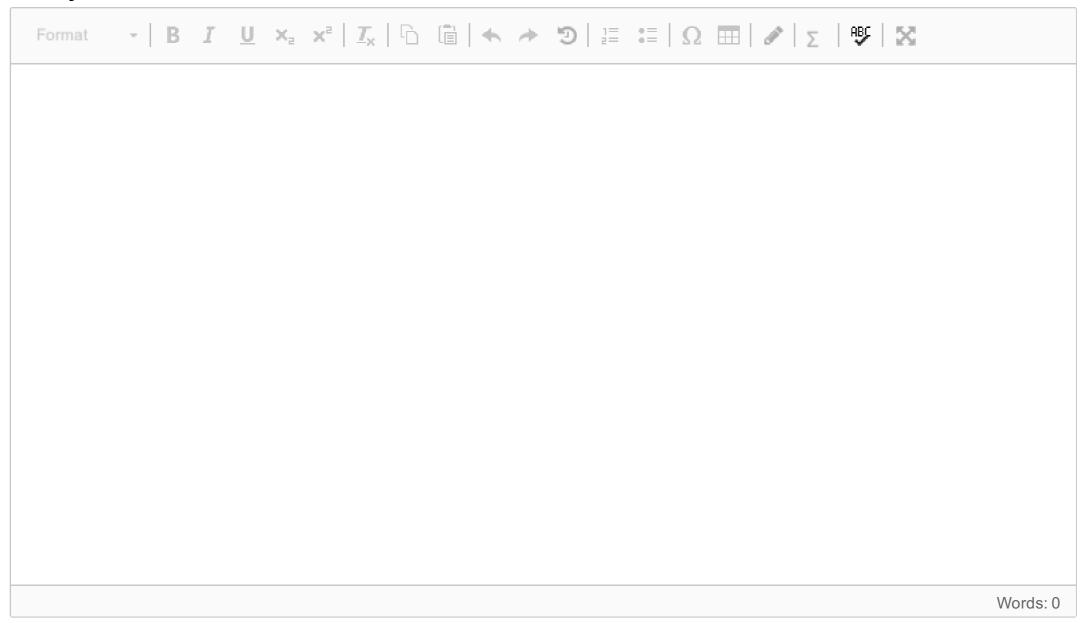
Fill in your answer here



Maximum marks: 4

18 myCPU

Write a DL-axiom asserting that *myCPU* is a CPU that is part of a super-computer.

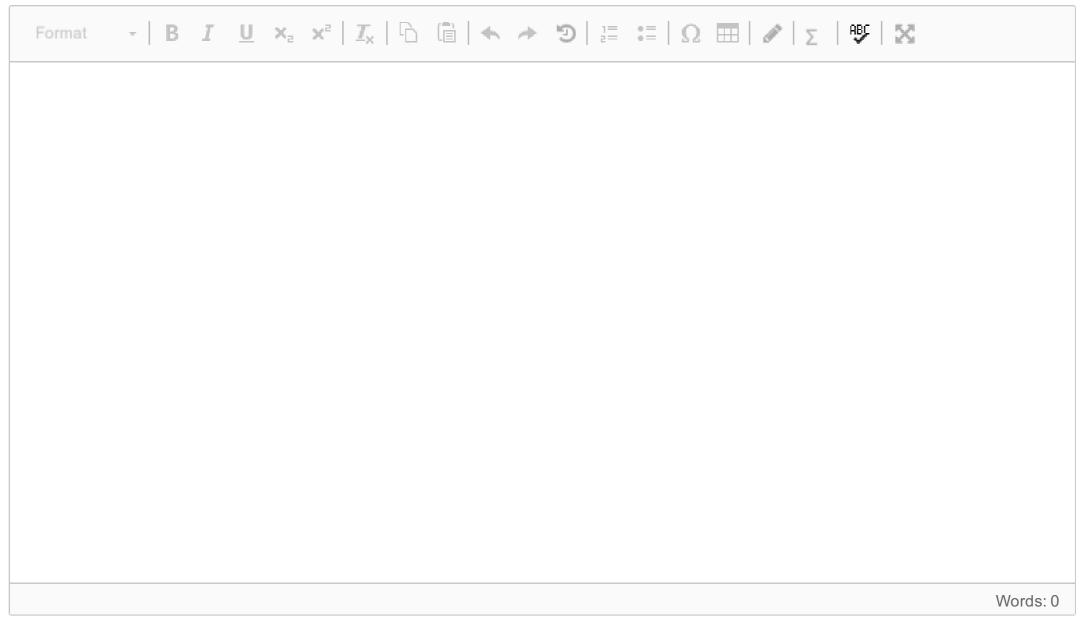


Maximum marks: 3

19 Connected to itself

Write a DL-axiom stating that nothing is connected to itself.

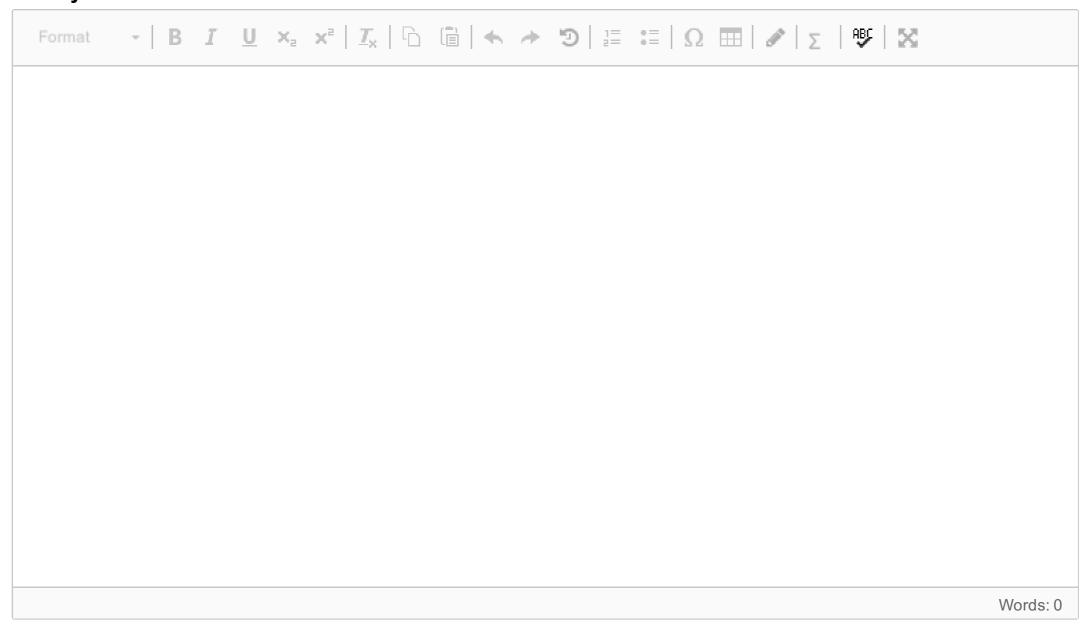
Fill in your answer here



Maximum marks: 3

Not having parts

Write a DL-axiom stating that if a thing $m{x}$ has a CPU with a core $m{y}$, then $m{x}$ has $m{y}$ as a core.



Maximum marks: 4

i Section 5: RDF and OWL semanticsSection 5: RDF and OWL semantics

For writing OWL or DL expressions, please use the Manchester syntax, see the table in Section 4.

21 Blank node semantics

Let I be the following interpretation:

```
Δ<sup>I</sup> = {robert, cersei, jaime, joffrey}

Man<sup>I</sup> = {robert, jaime, joffrey}

Woman<sup>I</sup> = {cersei}

King<sup>I</sup> = {robert}

marriedTo<sup>I</sup> = {<cersei,robert>}

hasFather<sup>I</sup> = {<joffrey,jaime>}

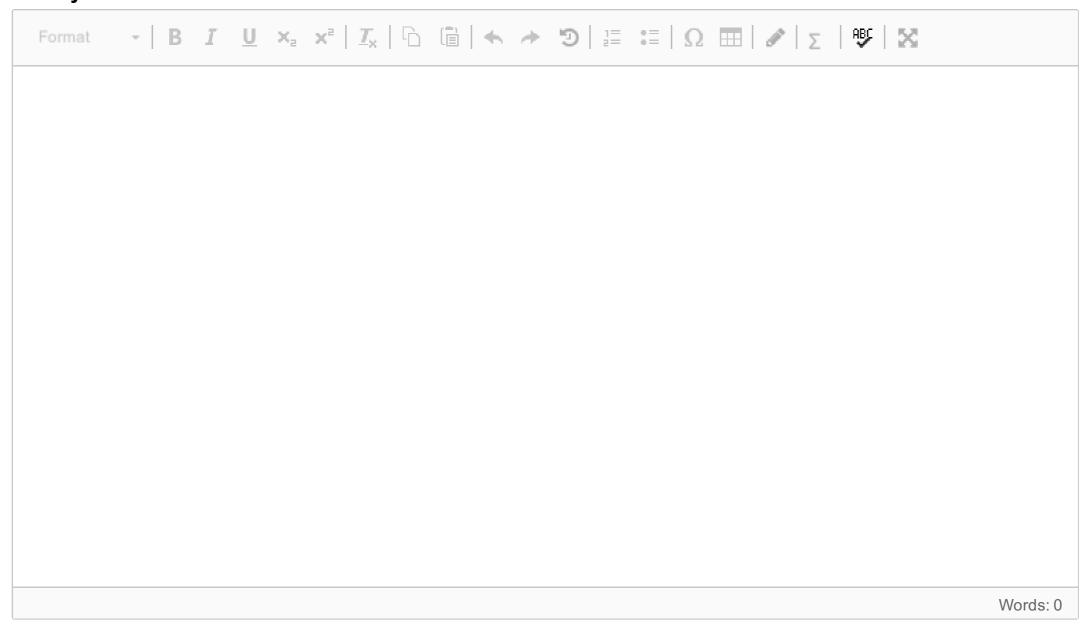
hasMother<sup>I</sup> = {<joffrey, cersei>}

Let A be the set of triples
{_:b :hasFather_:c, _:c rdf:type King }

or in DL notation:
{ hasFather(_:b,_:c), King(_:c) }
```

Is it the case that $I \models A$, i.e. A is true in I?

Please explain your answer, using the simplified RDFS model semantics from the course.



Maximum marks: 7

OWL Concept semantics

Let I be the same interpretation as before, repeated here for convenience.

```
\Delta^{l} = {robert, cersei, jaime, joffrey}

Man^{l} = {robert, jaime, joffrey}

Woman^{l} = {cersei}

King^{l} = {robert}

marriedTo^{l} = {<cersei,robert>, <robert,cersei>}

hasFather^{l} = {<joffrey,jaime>}

hasMother^{l} = {<joffrey, cersei>}

Please give the interpretation of the OWL concepts

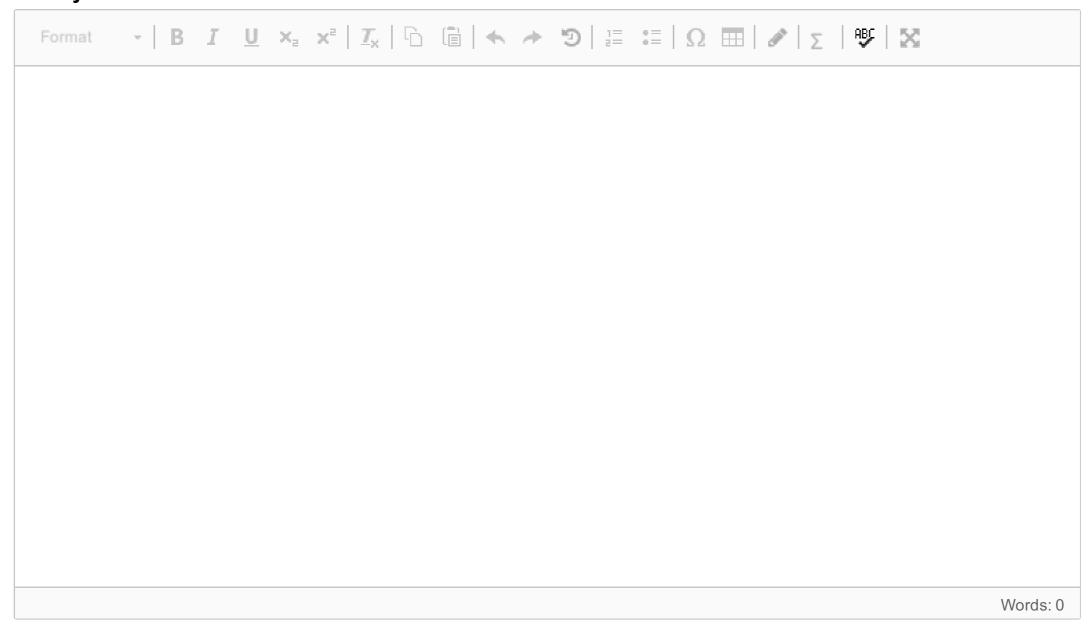
a) King or marriedTo some King

(In DL syntax: King \sqcup \exists marriedTo . King)

b) King or marriedTo only King

(In DL syntax: King \sqcup \forall marriedTo . King)
```

Please explain your answer, using the OWL model semantics from the course.



Maximum marks: 6

23 OWL Axiom semantics

Let I be the same interpretation as before, repeated here for convenience.

```
\Delta^{l} = {robert, cersei, jaime, joffrey}

Man^{l} = {robert, jaime, joffrey}

Woman^{l} = {cersei}

King^{l} = {robert}

marriedTo^{l} = {<cersei,robert>, <robert,cersei>}

hasFather^{l} = {<joffrey,jaime>}

hasMother^{l} = {<joffrey, cersei>}

Do the following OWL axioms hold in I?

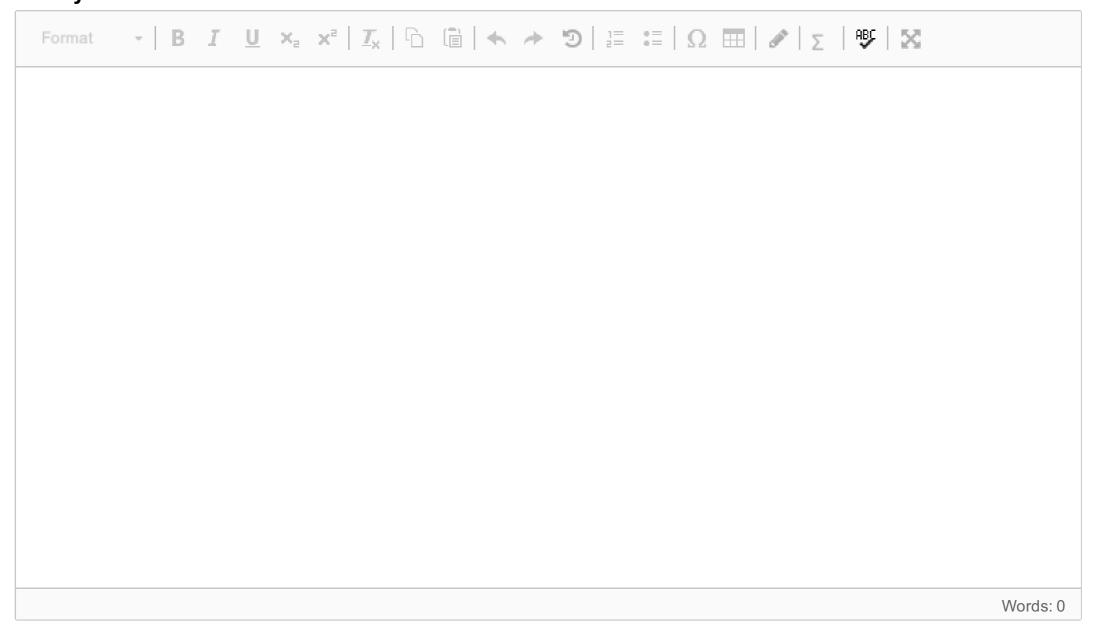
a) King subClassOf marriedTo only Woman

DL Syntax: King \sqsubseteq \forall marriedTo. Woman

b) King subClassOf marriedTo only owl:Thing

DL Syntax: King \sqsubseteq \forall marriedTo. \top
```

Is each of these an OWL tautology, i.e. does it hold in any interpretation?



Maximum marks: 7