Q8. Identify two different assertions that would make the ontology inconsistent.

Case 1.	**Case 2.**
bompiani :hasLegalName *Bompiani*	*someone* rdf:type :*Human*
bompiani :hasLegalName *Giunti Editore*	*someone* rdf:type :*NotHuman*
hasLegalName -> Functional	*Human* owl:disjointWith *NotHuman*

Q9. Define the complex role inclusion axiom capturing the fact that if a narrator creates a narrative that is reported in a book that is published by a publisher, then the narrator has a contract with that publisher.

V_C = {:Narrator, :Narrative, :Book, :Publisher}
V_{OP} = {:isCreatorOfNarrative, :isReportedByBook, :hasPublisher, :hasContractWith}

SubObjectPropertyOf(ObjectPropertyChain (:isCreatorOfNarrative, :isReportedByBook, :hasPublisher) :hasContractWith)

(isCreatorOfNarrative o isReportedByBook o hasPublisher) c hasContractWith

Please note the *hasContractWith* property has *Narrator* as (part of) its Domain and Range, but it would have made more sense to limit the property to *Living* narrators since dead people can't reasonably "have a contract" with someone. However, this would have been made the ontology inconsistent.

This is due to the fact that the restrictions of the Domain of the property itself are not taken into accounts when making an inference based on a SuperProperty Chain. Rather, the Domain of the sub-properties of the chains is used. Since the first property of the chain (*isCreatorOfNarrative*) has *Narrator* (including both *Living* and *Dead* narrators) as Domain, by setting *Living* as the Domain of *hasContractWith* we would have ended up with *Dead* narrators classified as both *Dead* and *Living* (and still as having a contract with a publisher, leaving us with the initial problem unsolved).

This is an example of the chains of inferences that would have been made:

- Only *Living* narrators have a contract with a publisher;
- Jonathan Swift is a *Dead* narrator;
- Jonathan Swift created the Gulliver narrative, which is reported by *Gulliver's Travels* book, which is published by Wordsworth Edition. Therefore, Jonathan Swift has a contract with Wordsworth Edition;
- As only *Living* narrators have a contract with a publisher, Jonathan Swift is *Living;*
- Jonathan Swift is both *Dead* and *Living,* but those are mutually exclusive classes as they are disjointed.

A possible solution to the initial problem could have been splitting *isCreatorOfNarrative* into two separated properties: *isTheLivingCreatorOfNarrative* and *isTheDeadCreatorOfNarrative,* with only the former included in the *hasContractWith* SuperProperty of Chains. We chose instead to define *hasContractWith* in a particular way when describing the property intension.

It should also be noted that, as *hasContractWith* is a symmetric property, defining the inverse of the chain (i.e.: isPublisherOf o reportsNarrative o narrativeCreatedBy) would have been redundant.

Q10. Verify if the created ontology (including the complex role inclusion axiom defined in Q9) satisfies the global restrictions on the axioms of an OWL 2 DL ontology.

Restriction on owl:topDataProperty: **Satisfied**, as *owl:topDataProperty* has no axioms, nor has any super properties.

Restrictions on Datatypes: Both are **satisfied**, as each datatype of the ontology is contained in the OWL 2 datatype map (and therefore they all follow a strict partitional order).

Restriction on Simple Roles: **Satisfied**, as no composite object property is used in one of the problematic axioms.

Restriction on the Property Hierarchy: **Satisfied**, because the five property chains we used in the ontology are all acyclic, as they all include simple properties (i.e., properties that are not defined by other properties):

- The chain of *hasContractWith* is made up by *isCreatorOfNarrative, isReportedByBook* and *hasPublisher;*
- The chain of *narrativeHasActant* is made up by *narrativeHasEvent* and *eventHasActant*;
- The chain of *isActantOfNarrative* is made up by *isActantOfEvent* and *isEventOfNarrative;*
- The chain of *narrativeHasLocation* is made up by *narrativeHasEvent* and *eventHasLocation;*
- The chain of *isLocationOfNarrative* is made up by *isLocationOfEvent* and *isEventOfNarrative.*

Restrictions on Anonymous Individuals: **Satisfied**, as we didn't use any anonymous individual in our ontology.

Q11. Write the following queries in SPARQL

Q11.1. Find how many events occurred in real locations, grouped by location.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://www.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>

SELECT (COUNT (?location) AS ?numberOfLocations) ?locationName

WHERE {
?location a :Real;
    :hasName ?locationName.
?event a dctypes:Event;
    :eventHasLocation ?location
}

GROUP BY ?locationName
```

numberOfLocations	locationName
1	Dublin
1	London

Q11.2. Find all the books with the ID of the publisher lower than 5000.

```
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.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>

SELECT ?bookLabel

WHERE {
?publisher a :Publisher;
    :hasID ?id .
?book a :Book;
    :hasPublisher ?publisher;
    rdfs:label ?bookLabel.

FILTER (?id < "5000"^^xsd:positiveInteger)
}
```

bookLabel
Book: Dubliners
Book: Gulliver's Travel
Book: Lo Hobbit
Book: Il Ritorno del Re (ISdA #3)

Q11.3. Find all the events that do not have any human participants.

```
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.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>

SELECT ?eventLabel

WHERE {
?event a dctypes:Event;
    :eventHasActant ?actant;
    rdfs:label ?eventLabel.

MINUS
{?actant a :Human}

}
```

eventLabel
Event (Good Omens): Learning about the Apocalypse
Event (The Hobbit): Bilbo gets the Ring
Event (The Hobbit): Bilbo gets the Ring
Event (The Hobbit): Bilbo gets the Ring
Event (ROTK): The Ring is destroyed

Note we are getting the same event listed multiple times – one time for each not-human actant (e.g. Bilbo, Gollum, the Ring). If we want to have a list of unique events we can use the SELECT DISTINCT command.

```
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.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>

SELECT DISTINCT ?eventLabel

WHERE {
?event a dctypes:Event;
    :eventHasActant ?actant;
    rdfs:label ?eventLabel.

MINUS
{?actant a :Human}

}
```

eventLabel
Event (Good Omens): Learning about the Apocalypse
Event (The Hobbit): Bilbo gets the Ring
Event (ROTK): The Ring is destroyed

Q11.4. Find the number of the narratives that are published in a book, along with the title of the book, the ISBN code of the book and the publisher of the book.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://www.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>

SELECT (COUNT (?narrative) AS ?numberOfNarratives) ?title ?publisherName ?isbn

WHERE {?narrative a :Narrative .
?book a :Book ;
    :reportsNarrative ?narrative;
    :hasTitle ?title;
    :hasPublisher ?publisher;
    :hasISBN ?isbn.
?publisher :hasLegalName ?publisherName.
}

GROUP BY ?title ?publisherName ?isbn
```

numberOfNarratives	title	publisherName	isbn
2	Dubliners	Wordsworth Editions Ltd	1853260487
1	Good Omens	HarperCollins Publishers Inc	0060853980
1	Gulliver's Travels	Wordsworth Editions Ltd	1853260274
1	Travels into Several Remote Nations of the World. In Four parts. By Lemuel Gulliver, First a Surgeon, and then a Captain of Several Ships	Wordsworth Editions Ltd	1853260274
1	The Hobbit: or There and Back Again	HarperCollins Publishers Inc	0261102214
1	Lo Hobbit. Un viaggio inaspettato	Bompiani - Giunti Editore	8845268349
1	Il Ritorno del Re	Bompiani - Giunti Editore	8830102725
1	The Return of the King	HarperCollins Publishers Inc	0261102370

Since *Gulliver's Travels* has multiple titles, it is listed multiple times. If we want to have it listed only once, we can use the SAMPLE command.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX : <http://www.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>

SELECT (COUNT (?narrative) AS ?numberOfNarratives)  (SAMPLE(?title) AS ?bookTitle) ?publisherName ?isbn

WHERE {?narrative a :Narrative .
?book a :Book ;
    :reportsNarrative ?narrative;
    :hasTitle ?title;
    :hasPublisher ?publisher;
    :hasISBN ?isbn.
?publisher :hasLegalName ?publisherName.
}

GROUP BY ?publisherName ?isbn
```

numberOfNarratives	bookTitle	publisherName	isbn
2	Dubliners	Wordsworth Editions Ltd	1853260487
1	Good Omens	HarperCollins Publishers Inc	0060853980
2	Gulliver's Travels	Wordsworth Editions Ltd	1853260274
1	The Hobbit: or There and Back Again	HarperCollins Publishers Inc	0261102214
1	Lo Hobbit. Un viaggio inaspettato	Bompiani - Giunti Editore	8845268349
1	Il Ritorno del Re	Bompiani - Giunti Editore	8830102725
1	The Return of the King	HarperCollins Publishers Inc	0261102370

Q11.5. Find all the distinct events that have a human participant or occur in a real location.

```
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.semanticweb.org/federico/ontologies/2021/1/narrative-ontology#>
PREFIX dctypes: <http://purl.org/dc/dcmitype/>

SELECT DISTINCT ?eventLabel


WHERE {
?actant a :Human.
?location a :Real.
?event a dctypes:Event;
    rdfs:label ?eventLabel.
{?event :eventHasActant ?actant.}
UNION
{?event :eventHasLocation ?location.}
}
```

eventLabel
Event (Eveline): Eveline's Choice
Event (Good Omens): Learning about the Apocalypse
Event (The Dead): Gabriel's Speech
Event (Gulliver's Travel): Lilliput
Event (Gulliver's Travel): Laputa
Event (Discworld): Unknown Event