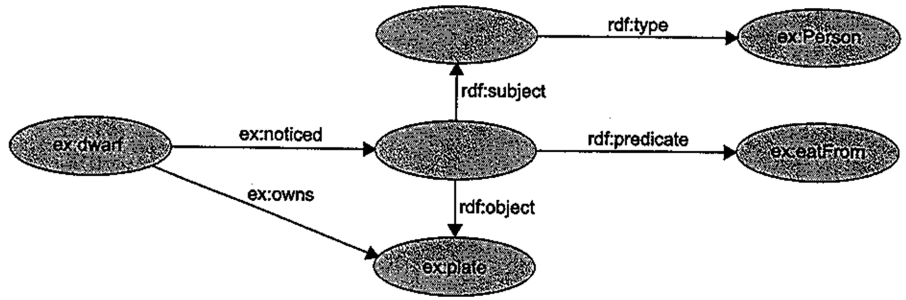

Solution to Exercise 2.5
The following sentences cannot be modeled satisfactorily in RDFS:

- Pizzas always have at least two toppings.
- Everything having a topping is a pizza.
- No pizza from the class `PizzaMargarita` has a topping from the class `Meat`.

The other sentences can be modeled as follows.

```
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex:   <http://example.org/> .

ex:Pizza            rdf:type         ex:Meal .
ex:PizzaMargarita   ex:hasTopping    ex:Tomato .
ex:hasTopping       rdf:type         rdfs:ContainerMembershipProperty .
```

D.2 Solutions for Chapter 3

Solution to Exercise 3.1
To make the solution really simple, set $IR = IP = \{a\}$ and $I_{EXT}(a) = \{\langle a, a\rangle\}$. Furthermore, I_S maps everything to a and $LV = I_L = \emptyset$.

Solution to Exercise 3.2
To solve this exercise, it is easiest to use the deduction rules from Section 3.3.

An example of a simple entailment is
`ex:vegetableThaiCurry ex:thaiDishBasedOn _:id1 .`

An example of an RDF-entailed triple which is not simply entailed is
`ex:thaiDishBasedOn rdf:type rdf:Property .`

An example of an RDFS-entailed triple which is not RDF-entailed is
`ex:vegetableThaiCurry rdf:type ex:Thai .`

Solution to Exercise 3.3
It is really not possible to specify this in RDFS.

Solution to Exercise 3.4
Using rdfsax, all the axiomatic triples listed for the RDF and RDFS semantics are derivable. We can now use the deduction rules for RDFS entailment to obtain further triples.

From `rdfs:domain rdfs:range rdfs:Class .`
and `rdf:type rdfs:domain rdfs:Resource .`
we can deduce using rdfs3 that
`rdfs:Resource rdf:type rdfs:Class .`

From `rdfs:range rdfs:range rdfs:Class .`
and `rdfs:range rdfs:range rdfs:Class .`
we can deduce using rdfs3 that
`rdfs:Class rdf:type rdfs:Class .`

From `rdfs:range rdfs:range rdfs:Class .`
and `rdfs:comment rdfs:range rdfs:Literal .`
we can deduce using rdfs3 that
`rdfs:Literal rdf:type rdfs:Class .`

From `rdfs:subClassOf rdfs:domain rdfs:Class .`
and `rdf:XMLLiteral rdfs:subClassOf rdfs:Literal .`
we can deduce using rdfs2 that
`rdfs:XMLLiteral rdf:type rdfs:Class .`

From `rdf:type rdfs:range rdfs:Class .`
and `rdf:XMLLiteral rdf:type rdfs:Datatype .`
we can deduce using rdfs3 that
`rdfs:Datatype rdf:type rdfs:Class .`

From `rdfs:subClassOf rdfs:domain rdfs:Class .`
and `rdf:Seq rdfs:subClassOf rdfs:Container .`
we can deduce using rdfs2 that
`rdfs:Seq rdf:type rdfs:Class .`

From `rdfs:subClassOf rdfs:domain rdfs:Class .`
and `rdf:Bag rdfs:subClassOf rdfs:Container .`
we can deduce using rdfs2 that
`rdfs:Seq rdf:type rdfs:Class .`

From `rdfs:subClassOf rdfs:domain rdfs:Class .`
and `rdf:Alt rdfs:subClassOf rdfs:Container .`
we can deduce using rdfs2 that
`rdfs:Seq rdf:type rdfs:Class .`

From `rdfs:subClassOf rdfs:range rdfs:Class .`
and `rdf:Alt rdfs:subClassOf rdfs:Container .`
we can deduce using rdfs3 that
`rdfs:Container rdf:type rdfs:Class .`

From `rdfs:domain rdfs:range rdfs:Class .`
and `rdf:first rdfs:domain rdfs:List .`
we can deduce using rdfs3 that
`rdfs:List rdf:type rdfs:Class .`

From `rdfs:subClassOf rdfs:domain rdfs:Class .`
and `rdf:ContainerMembershipProperty rdfs:subClassOf rdfs:Property .`
we can deduce using rdfs2 that
`rdf:ContainerMembershipProperty rdf:type rdfs:Class .`

From `rdfs:range rdfs:range rdfs:Class .`
and `rdfs:subPropertyOf rdfs:range rdf:Property .`
we can deduce using rdfs3 that
`rdf:Property rdf:type rdfs:Class .`

From `rdfs:domain rdfs:range rdfs:Class .`
and `rdf:subject rdfs:domain rdf:Statement .`
we can deduce using rdfs3 that
`rdf:Statement rdf:type rdfs:Class .`

From `rdfs:range rdfs:domain rdf:Property .`
we can deduce using rdf1 that
`rdfs:domain rdf:type rdf:Property .`

From `rdfs:subPropertyOf rdfs:range rdf:Property .`
we can deduce using rdf1 that
`rdfs:range rdf:type rdf:Property .`

From `rdfs:isDefinedBy rdfs:subPropertyOf rdfs:seeAlso .`
we can deduce using rdf1 that
`rdfs:subPropertyOf rdf:type rdf:Property .`

From `rdf:Alt rdfs:subClassOf rdfs:Container .`
we can deduce using rdf1 that
`rdfs:subClassOf rdf:type rdf:Property .`

From `rdf:Alt rdfs:subClassOf rdfs:Container .`
we can deduce using rdf1 that
`rdfs:subClassOf rdf:type rdf:Property .`

From `rdfs:range rdfs:domain rdf:Property .`
and `rdfs:member rdfs:range rdfs:Resource .`
we can deduce using rdfs2 that
`rdfs:member rdf:type rdfs:Property .`

From `rdfs:range rdfs:domain rdf:Property .`
and `rdfs:seeAlso rdfs:range rdfs:Resource .`
we can deduce using rdfs2 that
`rdfs:seeAlso rdf:type rdfs:Property .`

From `rdfs:range rdfs:domain rdf:Property .`
and `rdfs:isDefinedBy rdfs:range rdfs:Resource .`
we can deduce using rdfs2 that
`rdfs:isDefinedBy rdf:type rdfs:Property .`

From `rdfs:range rdfs:domain rdf:Property .`
and `rdfs:comment rdfs:range rdfs:Literal .`
we can deduce using rdfs2 that
`rdfs:comment rdf:type rdfs:Property .`

From `rdfs:range rdfs:domain rdf:Property .`
and `rdfs:label rdfs:range rdfs:Literal .`
we can deduce using rdfs2 that
`rdfs:label rdf:type rdfs:Property .`

D.3 Solutions for Chapter 4

Solution to Exercise 4.1

```
<owl:Class rdf:about="Vegetable">
  <rdfs:subClassOf rdf:resource="PizzaTopping" />
</owl:Class>
<owl:Class rdf:about="PizzaTopping">
  <rdfs:disjointWith rdf:resource="Pizza" />
<Vegetable rdf:about="aubergine" />
<owl:ObjectProperty rdf:about="hasTopping">
  <rdfs:domain rdf:resource="Pizza" />
```