Exercise 3.1. Model the following statements in OWL DL by giving the corresponding OWL/RDF snippets:

- The class Vegetable is a subclass of PizzaTopping.
- The class PizzaTopping does not have common elements with the class Pizza.
- The individual aubergine is an element of the class Vegetable.
- The abstract role hasTopping connects only elements of the class Pizza with elements of the class PizzaTopping.
- Pizzas always have at least two toppings.
- Every Pizza from the class PizzaMargarita has Tomato as topping.
- The class VegetarianPizza consists of those individuals that are both in the class PizzaWithoutMeat and in the class PizzaWithoutFish.
- No Pizza from the class PizzaMargarita has a topping from the class Meat.

Exercise 3.2. Decide, if the following statements would make sense in the context of the pizza ontology from Exercise 3.1:

- The role hasIngredient is transitive.
- The role hasTopping is functional.
- The role hasTopping is inverse functional.
- The role hasIngredient is asymmetric.

Exercise 3.3. Assume a vocabulary with the individual names bonny and clyde, the class names Honest, Wise, Crime and Human as well as the role names commits, marriedWith, suspects, report and know.

Which of the following propositions can be made in OWL 1, which in OWL 2 and which ones not at all? In the positive case, provide the corresponding axioms.
(a) Everybody, who is honest and who commits a crime, reports himself.
(b) Who is wise and honest, doesn’t commit crimes.
(c) Bonnie does not report Clyde.
(d) Nobody reports a human, which whom he has committed a crime jointly.
(e) Clyde has committed at least 10 crimes.
(f) Bonnie and Clyde have committed at least one crime together.
(g) Who committed a crime together with his/her spouse is not honest.
(h) Everybody knowing a suspect, is a suspect himself.

Exercise 3.4. We want to define the concept “vegetarian pizza”. Which of the following definitions are appropriate for this? Provide a natural language description for each of the logical statements.

(a) VegetarianPizza ≡ Pizza ⊓¬∃hasIngredient.(Meat ∪ Fish)
(b) VegetarianPizza ≡ Pizza ⊓ ∀hasTopping.(¬Meat ∪ ¬Fish)
(c) VegetarianPizza ≡ Pizza ⊓¬∃hasTopping.Meat ⊓¬∃hasTopping.Fish
(d) VegetarianPizza ≡ Pizza ⊓∃hasTopping.¬Meat ⊓∃hasTopping.¬Fish
(e) VegetarianPizza ≡ Pizza ⊓∀hasIngredient.¬Meat ∪ ¬Fish

Exercise 3.5. Let the following ontology in DL syntax be given:

\[
\begin{align*}
\text{hasTopping} & \sqsubseteq \text{hasIngredient} \\
\text{Vegetable} & \sqcap \text{Cheese} \sqsubseteq \bot \\
\text{Vegetable} & \sqcap \text{Meat} \sqsubseteq \bot \\
\text{Vegetable} & \sqcap \text{Fish} \sqsubseteq \bot \\
\top & \sqsubseteq \forall \text{hasTopping}.\text{PizzaTopping}
\end{align*}
\]

Consider the following additional class definitions:

\[
\begin{align*}
\text{CheesePizza} & \equiv \text{Pizza} \sqcap \exists \text{hasTopping}.\text{Cheese} \\
\text{PizzaSpinach} & \equiv \exists \text{hasTopping}.\text{Spinach} \sqcap \exists \text{hasTopping}.\text{Cheese} \\
& \quad \sqcap \forall \text{hasTopping}.(\text{Spinach} \sqcup \text{Cheese}) \\
\text{PizzaCarnivorus} & \equiv \text{Pizza} \sqcap \neg \exists \text{hasTopping}.(\text{Meat} \sqcup \text{Fish}) \\
\text{EmptyPizza} & \equiv \text{Pizza} \sqcap \neg \exists \text{hasTopping}.\top
\end{align*}
\]

(a) Which of the classes given above would be recognized as subclass of VegetarianPizza by a DL reasoner (according to a correct definition from Exercise 3.4? Explain your answers.
(b) The classification from (a) shows that some of the pizza classes do not model the intended concept. How could the definition be corrected?

(c) How would the result from (a) change, if one would use just $\subseteq$ instead of $\equiv$ when defining VegetarianPizza?