# KRP — Assignment 5

Instructor: Yizheng Zhao

May 28, 2024

⋆ **This assignment, due on 30th June at 23:59, contributes to 10% of the final marks for this course. Please be advised that only Questions 1 — 5 are mandatory. Nevertheless, students can earn up to one bonus mark by completing Question 6. This bonus mark can potentially augment a student's overall marks but is subject to a maximum total of 100 for the course. By providing bonus marks, we aim to incentivize students to excel in their studies and reward those with a remarkable grasp of the course materials.**

**Question 1. Ontology-Mediated Query Answering (2 marks)**

(1) Consider the database instance $\mathcal{D}_{\text{PGBBW}}$ (PGBBW stands for Pleasant Goat and Big Big Wolf) below:

Sheep(weslie) Sheep(slowy) LazySheep(paddi)

Sheep(tibbie) BrownSheep(fitty) Sheep(jonie)

Wolf(wolffy) Wolf(wolnie) Wolf(wilie)

hasFriend(weslie, slowy) hasFriend(tibbie, jonie)

hasEnemy(paddi, wolffy) hasWife(wolffy, wolnie) hasSon(wolnie, wilie)

We query $\mathcal{D}_{\text{PGBBW}}$ under closed world assumption (relational database) and under open world assumption. Recall that under the closed world assumption we consider the interpretation $\mathcal{I} := \mathcal{I}_{\mathcal{D}_{\text{PGBBW}}}$ defined as follows:

$\Delta^{\mathcal{I}} = \{\text{weslie, slowy, paddi, tibbie, fitty, jonie, wolffy, wolnie, wilie}\}$

$\text{Sheep}^{\mathcal{I}} = \{\text{weslie, slowy, tibbie, jonie}\}$

$\mathsf{LazySheep}^{\mathcal{I}} = \{\mathsf{paddi}\}$

$\mathsf{BrownSheep}^{\mathcal{I}} = \{\mathsf{fitty}\}$

$\mathsf{Wolf}^{\mathcal{I}} = \{\mathsf{wolffy}, \mathsf{wolnie}, \mathsf{wilie}\}$

$\mathsf{hasFriend}^{\mathcal{I}} = \{(\mathsf{weslie}, \mathsf{slowy}), (\mathsf{tibbie}, \mathsf{jonie})\}$

$\mathsf{hasEnemy}^{\mathcal{I}} = \{(\mathsf{paddi}, \mathsf{wolffy})\}$

$\mathsf{hasWife}^{\mathcal{I}} = \{(\mathsf{wolffy}, \mathsf{wolnie})\}$

$\mathsf{hasSon}^{\mathcal{I}} = \{(\mathsf{wolnie}, \mathsf{wilie})\}$

Consider the following non-Boolean queries $F_i$:

1. $F_1(x) = \mathsf{Wolf}(x)$
2. $F_2(x) = \neg\mathsf{Sheep}(x)$
3. $F_3(x, y) = \mathsf{hasFriend}(x, y)$
4. $F_4(x) = \mathsf{Sheep}(x) \wedge \neg\mathsf{hasFriend}(x, \mathsf{jonie})$

For each $F_i$, give

- for closed world assumption: answer($F_i$,$\mathcal{D}_{\mathsf{PGBBW}}$);
- for open world assumption: certanswer($F_i$,$\mathcal{D}_{\mathsf{PGBBW}}$).

(2). Following (1), consider now the TBox $\mathcal{T}$ given as:

$$\begin{aligned} \mathsf{LazySheep} &\sqsubseteq \mathsf{Sheep} \\ \mathsf{LazySheep} \sqcap \mathsf{BrownSheep} &\sqsubseteq \bot \\ \mathsf{Sheep} \sqcap \mathsf{Wolf} &\sqsubseteq \bot \\ \top &\sqsubseteq \forall\mathsf{hasFriend}.\mathsf{Sheep} \\ \exists\mathsf{hasFriend}.\mathsf{Sheep} &\sqsubseteq \mathsf{Sheep} \\ \mathsf{Sheep} &\sqsubseteq \exists\mathsf{hasFriend}.\top \end{aligned}$$

Fill in the table below with the answers “Yes”, “No” or “Don’t know” to the Boolean queries.

| Query | Answer for $\mathcal{I}$ | Certain Answer for $\mathcal{D}_{\mathsf{PGBBW}}$ | Certain Answer for $(\mathcal{T}, \mathcal{D}_{\mathsf{PGBBW}})$ |
|---|---|---|---|
| LazySheep(paddi) | | | |
| LazySheep(fitty) | | | |
| Sheep(paddi) | | | |
| ¬Sheep(paddi) | | | |
| BrownSheep(wilie) | | | |
| Wolf(fitty) | | | |
| ∃hasFriend.⊤(fitty) | | | |
| ∀hasFriend.⊤(wolffy) | | | |
| ∀hasFriend.∃hasFriend.⊤(wolnie) | | | |
| ∃hasFriend.∀hasFriend.⊤(jonie) | | | |

**Question 2. Computing $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ in $\mathcal{EL}$**

Consider the following $\mathcal{EL}$ TBox $\mathcal{T}$:

$$\begin{aligned}
\text{Vocalist} &\sqsubseteq \exists \text{plays_for.Band} \\
\text{Guitarist} &\sqsubseteq \exists \text{plays_for.Band} \\
\text{Bassist} &\sqsubseteq \exists \text{plays_for.Band} \\
\text{Drummer} &\sqsubseteq \exists \text{plays_for.Band} \\
\text{LeadGuitarist} &\sqsubseteq \text{Guitarist} \\
\text{RhythmGuitarist} &\sqsubseteq \text{Guitarist} \\
\text{Band} &\sqsubseteq \exists \text{captained_by.Captain} \\
\text{Band} &\sqsubseteq \exists \text{managed_by.Manager} \\
\text{Manager} &\sqsubseteq \exists \text{managed_by.Manager}
\end{aligned}$$

and the following ABox $\mathcal{A}$:

| | |
|---|---|
| Captain(Monster) | Vocalist(Ashin) |
| LeadGuitarist(Monster) | Bassist(Masa) |
| RhythmGuitarist(Stone) | Drummer(Ming) |
| Band(Mayday) | managed_by(Mayday, Amuse) |

- Compute the interpretation $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ as described in the slides.
- For $\mathcal{EL}$ concept queries, we know that $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ gives the answer "Yes" iff $(\mathcal{T}, \mathcal{A})$ gives the certain answer "Yes". Check this for the following queries:
  1. $\exists$plays_for.Band(Ashin);
  2. $\exists$managed_by.Manager(Masa);
  3. $\exists$plays_for.$\exists$captained_by.Captain(Monster);

4. $\exists$plays_for.$\exists$managed_by.Manager(Ming).

- For more complex queries, $\mathcal{I}_{\mathcal{T},\mathcal{A}}$ can give the answer "Yes" even if $(\mathcal{T}, \mathcal{A})$ does not give the certain answer "Yes". Check this for:

1. $F(x, y) = \exists z.(\text{plays\_for}(x, z) \wedge \text{plays\_for}(y, z))$.
2. $F = \exists x.\text{managed\_by}(x, x)$.

**Question 3. SQL and Conjunctive Queries (2 marks)**
Consider the following database $\mathcal{D}$ consisting of the following tables:

| Person: | |
|---|---|
| ID | Name |
| 2101 | Jay_Chou |
| 2102 | Jolin_Tsai |
| 2103 | Stefanie_Sun |
| 2104 | Ta-yu_Lo |

| Enrollment: | |
|---|---|
| StudentID | Since |
| 2102 | 2020 |
| 2103 | 2021 |
| 2104 | 2020 |

| Attendance: | |
|---|---|
| StudentID | CourseID |
| 2101 | 30000160 |
| 2102 | 30000160 |
| 2102 | 30000170 |
| 2103 | 30000150 |

| Course: | |
|---|---|
| ID | Title |
| 30000150 | ML |
| 30000160 | KRP |
| 30000170 | NLP |

- Define the finite first-order interpretation $\mathcal{I}_{\mathcal{D}}$ corresponding to $\mathcal{D}$.

- Reformulate each of the following SQL queries $Q$ into first-order queries $f_Q$, and identify which of them are conjunctive queries.

- Answer $Q$ in the context of $\mathcal{D}$ and $f_Q$ in the context of $\mathcal{I}_{\mathcal{D}}$.

1. SELECT $*$ FROM Person
2. SELECT Person.Name FROM Person, Attendance, Course
   WHERE Person.ID = Attendance.PersonID
   AND Course.ID = Attendance.CourseID
   AND Course.Title = "KRP"
3. SELECT Person.Name FROM Person, Enrollment
   WHERE Person.ID = Enrollment.PersonID
   AND NOT EXISTS (
   SELECT $*$ FROM Attendance
   WHERE Person.ID = Attendance.PersonID)

**Question 4. Certain Answers in $\mathcal{ALC}$ Contexts (2 marks)**
Consider the following $\mathcal{ALC}$ knowledge base $\mathcal{K} := (\mathcal{T}, \mathcal{A})$ with:

$$\mathcal{T} := \{X \sqsubseteq Y, Y \sqsubseteq \exists r.X, X \sqsubseteq \forall r.Y, \forall r.X \sqsubseteq Y, W \equiv \neg V, \exists r.Y \sqsubseteq \neg V\}$$
$$\mathcal{A} := \{(\text{Jay\_Chou}, \text{Jolin\_Tsai}) : r, (\text{Jolin\_Tsai}, \text{Stefanie\_Sun}) : r, (\text{Stefanie\_Sun}, \text{Jay\_Chou}) : r,$$
$$(\text{Jolin\_Tsai}, \text{Jolin\_Tsai}) : r, (\text{Stefanie\_Sun}, \text{Stefanie\_Sun}) : r, \text{Stefanie\_Sun} : X\}$$

- Compute the certain answers to the following conjunctive queries in the context of $\mathcal{A}$.

- Compute the certain answers to the following conjunctive queries in the context of $\mathcal{K}$.

1. $r(x, y) \wedge Y(y)$
2. $\exists y(r(x, y) \wedge Y(y))$

3. $\exists x, y(r(x, y) \wedge r(y, x))$
4. $\exists z, w(r(x, y) \wedge r(y, z) \wedge r(z, x) \wedge r(z, w) \wedge W(w))$

**Question 5. CW5: Implementing Ontology Queries (2 marks)**

CW5 focuses on implementing an OWL-based querying system. Your primary task involves querying the reasoner for sub-classes, equivalent classes, and instances, then storing these results in a QueryResult object. Within the Java class CW5, you are expected to implement specific methods. Please ensure you do not modify any classes other than CW5, as this is the only class you may submit. To evaluate your implementation, you can utilize the provided classes, App.java and PizzaOrderingSystemApp.java. These classes provide a user-friendly way to verify your code's correctness. You can run these classes in Eclipse (or your preferred IDE) by clicking the small white and green arrow icon when the class is open. For a more direct approach, consider using the pre-implemented unit tests to gauge your progress. Note that passing these unit tests does not guarantee a perfect score, as additional tests will be used to assess your solution more comprehensively. Upon completing the assignment, please compress your CW5.java into a zip file for submission. Good luck!

1. public Set<QueryResult> performQuery (OWLClassExpression exp, Query Type type) {...}
   This method takes in an arbitrary expression in OWL, like "Person and hasBirthYear value 1964" or "hasTopping some MeatTopping" and queries the ontology for the following three types of knowledge:
   EquivalentClasses: you are asked to return all those (named) classes that are equivalent to the expression (exp), using the (already fully initialized) reasoner.
   Sub-classes: return all those classes that are (named) sub-classes of the expression, using the reasoner.
   Instances: return all those individuals that are instances of the expression, using the reasoner.
   Query results are stored in query result objects. These are created for example as follows:
   QueryResult qr = new QueryResult(ind, false, type);
   Note that you need to include the information whether the inference is direct or indirect. Example: Given three classes with A subclass B subclass C, then A is a direct subclass of B and B is a direct subclass of C, but A is an indirect subclass of C (through B!). In order to solve this problem, you will query the reasoner for direct and indirect sub-class separately. After creating the QueryResult, add it to the provided set.

2. public Boolean isValidPizza (OWLClassExpression exp) {...};
   In this method, you check whether the supplied class expression exp is a valid pizza expression, i.e., whether it is inferred to be a Pizza.

3. public Set<QueryResult> filterNamedPizzas (Set<QueryResult> results) {...};
   This question is similar to the one before, only that you are asked to filter from a set of results those that correspond to NamedPizza's, such as Margherita or AmericanHot.

4. public Set<OWLClassExpression> getAllSuperClassExpressions (OWLClass ce) {...};

This question requires a bit of thinking. You are asked to query the ontology to gather all available information about the class ce. In order to get an idea of what "AL" means, you can open Protégé and look at the "Classe" tab. Clicking on a class will reveal its super-classes, equivalent-classes, as well as inherited super-classes (Anonymous Ancestors) and indirect super-classes. Unfortunately, the reasoner will not easily give you access to those. In order to get full marks for this task, it is sufficient to query for all super-classes (direct and indirect, using the reasoner), and somehow find a way to obtain the anonymous super-classes using the ontology directly (that will need a bit of thought, do not despair too quickly). A perfect solution will test, for all sub-expressions in the ontology, whether ce is a sub-concept of it. Attempt this only if you are really confident with the OWL API by now.

**Question 6 (with 1 bonus mark). Conservative Extension-based Module Extraction**

The design, development, maintenance, reuse, and integration of ontologies are complex tasks. Like software engineers, ontology engineers need to be supported by tools and methodologies that help them to minimize the introduction of errors, i.e., to ensure that ontologies are consistent and do not have unexpected logical consequences. In order to develop this support, important notions from the field of software engineering, such as *module*, *black-box behavior*, and *controlled interaction*, need to be adapted.

For example, consider a scenario where an ontology engineer is building an ontology about research projects, which specifies different types of projects according to the research topic they focus on. While the ontology engineer is an expert on research projects—(s)he knows, for example, that an EUProject is a Project (axiom P3)—(s)he may not be well-versed in the specific topics covered by the projects, particularly when it comes to terms such as "Cystic_Fibrosis" and "Genetic_Disorder" mentioned in axioms P1 and P2. To address this knowledge gap, the engineer decides to leverage the knowledge available in a well-established and widely-used medical ontology $\mathcal{Q}$ containing the axioms M1—M5 in Figure 1, where the concepts Cystic_Fibrosisis and Genetic_Disorder are described.

| | **Ontology of medical research projects $\mathcal{P}$:** |
|---|---|
| P1 | Genetic_Disorder_Project $\equiv$ Project $\sqcap \exists$has_Focus.Genetic_Disorder |
| P2 | Cystic_Fibrosis_EUProject $\equiv$ EUProject $\sqcap \exists$has_Focus.Cystic_Fibrosis |
| P3 | EUProject $\sqsubseteq$ Project |
| P4 | $\exists$has_Focus.$\top \sqsubseteq$ Project |
| E1 | Project $\sqcap$ (Genetic_Disorder $\sqcap$ Cystic_Fibrosis) $\sqsubseteq \bot$ |
| E2 | $\forall$has_Focus.Cystic_Fibrosis $\sqsubseteq \exists$has_Focus.Genetic_Disorder |
| | **Ontology of medical terms $\mathcal{Q}$:** |
| M1 | Cystic_Fibrosis $\equiv$ Fibrosis $\sqcap \exists$located_In.Pancreas $\sqcap \exists$has_Origin.Genetic_Origin |
| M2 | Genetic_Fibrosis $\equiv$ Fibrosis $\sqcap \exists$has_Origin.Genetic_Origin |
| M3 | Fibrosis $\sqcap \exists$located_In.Pancreas $\sqsubseteq$ Genetic_Fibrosis |
| M4 | Genetic_Fibrosis $\sqsubseteq$ Genetic_Disorder |
| M5 | DEFBI_Gene $\sqsubseteq$ Immuno_Protein_Gene $\sqcap \exists$associated_With.Cystic_Fibrosis |

Figure 1: Reusing medical terminology in an ontology on research projects

The most straightforward way to reuse these concepts is to import into $\mathcal{P}$ the ontology $\mathcal{Q}$—that is, to add the axioms from $\mathcal{Q}$ into $\mathcal{P}$ and work with the extended ontology $\mathcal{P} \cup \mathcal{Q}$. Importing additional axioms into an ontology may result in new logical consequences. For example, it is easy to see that axioms M1—M4 in $\mathcal{Q}$ imply that every instance of Cystic_Fibrosis is an instance of Genetic_Disorder:

$$\mathcal{Q} \models \alpha := \text{Cystic\_Fibrosis} \sqsubseteq \text{Genetic\_Disorder} \tag{1}$$

Indeed, the axiom

$$\alpha_1 := \text{Cystic\_Fibrosis} \sqsubseteq \text{Genetic\_Fibrosis} \tag{2}$$

follows from the axioms M1 and M2 as well as from the axioms M1 and M3; $\alpha$ follows from the axioms $\alpha_1$ and M4. Using $\alpha$ as well as the axioms P1—P3 in the ontology $\mathcal{P}$ we can now prove that every instance of Cystic_Fibrosis_EUProject is also an instance of Genetic_Disorder_Project:

$$\mathcal{P} \cup \mathcal{Q} \models \beta := \text{Cystic\_Fibrosis\_EUProject} \sqsubseteq \text{Genetic\_Disorder\_Project} \tag{3}$$

This axiom $\beta$, however, does not follow from $\mathcal{P}$ alone:

$$\mathcal{P} \not\models \beta := \text{Cystic\_Fibrosis\_EUProject} \sqsubseteq \text{Genetic\_Disorder\_Project} \tag{4}$$

The ontology engineer might not be aware of Entailment (3), even though it concerns the terms of primary scope in his/her projects ontology $\mathcal{P}$.

It could be expected that entailments like $\alpha$ in (1) from an imported ontology $\mathcal{Q}$ result in new logical consequences, like $\beta$ in (3), over the terms defined in the main ontology $\mathcal{P}$. However, it is not expected that the meaning of the terms defined in $\mathcal{Q}$ changes as a consequence of the import since these terms are supposed to be completely specified within $\mathcal{Q}$. Such a side effect would be highly undesirable for the modeling of $\mathcal{P}$ since the ontology engineer of $\mathcal{P}$ might not be an expert on the subject of $\mathcal{Q}$ and is not supposed to alter the meaning of the terms defined in $\mathcal{Q}$ not even implicitly. Nevertheless, it is possible that the meaning of the reused terms could inadvertently change during the import process, potentially due to modeling errors.

In order to illustrate such a situation, suppose that the ontology engineer has learned about the concepts Cystic_Fibrosis and Genetic_Disorder from the ontology $\mathcal{Q}$ (including the axiom $\alpha$ and has decided to introduce additional axioms formalizing the following statements:

*"Every instance of Project is different from every instance of Genetic_Disorder and every instance of Cystic_Fibrosis"* (5)

*"Every project that has_Focus on Cystic_Fibrosis, also has_Focus on Genetic_Disorder"* (6)

Note that the statements (5) and (6) can be thought of as adding new information about projects and, intuitively, they should not change or constrain the meaning of the medical terms. Suppose that the ontology engineer has formalized the statements (5) and (6) in ontology using axioms E1 and E2 respectively. At this point, the ontology engineer has introduced modeling errors and, as a consequence, axioms E1 and E2 do not correspond to (5) and (6). E1 actually formalizes the following statement:

*"Every instance of Project is different from every common instance of Genetic_Disorder and Cystic_Fibrosis"* (7)

and E2 expresses that:

*"Every project that either has_Focus on nothing or has_Focus on Cystic_Fibrosis, also has_Focus on Genetic_Disorder"* (8)

These kinds of modeling errors are difficult to detect, especially when they do not cause inconsistencies in the ontology. Note that, while axiom E1 does not correspond to (5), it is still a consequence of (5), which means that it should not constrain the meaning of the medical terms. On the other hand, E2 is not a consequence of (6), and, in fact, it constrains the meaning of medical terms. Indeed, the axioms E1 and E2 together with axioms P1—P4 from $\mathcal{P}$ imply new axioms about the concepts Cystic_Fibrosis and Genetic_Disorder, namely their disjointness:

$$\mathcal{P} \models \gamma := \mathsf{Cystic\_Fibrosis} \sqcap \mathsf{Genetic\_Disorder} \sqsubseteq \bot \tag{9}$$

The entailment (9) can be proved using axiom E2 which is equivalent to:

$$\top \sqsubseteq \exists \mathsf{has\_Focus}.(\neg \mathsf{Cystic\_Fibrosis} \sqcup \mathsf{Genetic\_Disorder}) \tag{10}$$

The inclusion (10) and P4 imply that every element in the domain must be a project—that is, $\mathcal{P} \models \top \sqsubseteq \mathsf{Project}$). Now, together with axiom E1, this implies (9). The axioms E1 and E2 not only imply new statements about the medical terms, but also cause inconsistencies when used together with the imported axioms from $\mathcal{Q}$. Indeed, from (1) and (9) we obtain:

$$\mathcal{P} \cup \mathcal{Q} \models \delta := \mathsf{Cystic\_Fibrosis} \sqsubseteq \bot \tag{11}$$

which expresses the unsatisfiability of the concept Cystic _Fibrosis.

In conclusion, it has been observed that importing an external ontology can lead to undesirable side effects in our knowledge reuse scenario. These consequences may appear as the entailment of new axioms or even the occurrence of unsatisfiability issues involving the reused vocabulary.

Thus, an important requirement for the reuse of an ontology $\mathcal{Q}$ within an ontology $\mathcal{P}$ should be that $\mathcal{P} \cup \mathcal{Q}$ produces exactly the same logical consequences over the vocabulary of $\mathcal{Q}$ as $\mathcal{Q}$ alone does. This requirement can be naturally formulated using the well-known notion of a conservative extension, which has recently been introduced in our lectures.

**Definition 1 (Deductive Conservative Extension)** *Let $\mathcal{T}_1$ and $\mathcal{T}_1 \subseteq \mathcal{T}_2$ be two $\mathcal{L}$-TBoxes and $\Sigma$ a signature over the description logic $\mathcal{L}$. We say that $\mathcal{T}_2$ is a* deductive $\Sigma$-conservative extension *of $\mathcal{T}_1$ w.r.t. $\mathcal{L}$, if for every $\mathcal{L}$-axiom $\alpha$ with $sig(\alpha) \subseteq \Sigma$, we have $\mathcal{T}_1 \models \alpha$ iff $\mathcal{T}_2 \models \alpha$. We say that $\mathcal{T}_2$ is a deductive conservative extension of $\mathcal{T}_1$ if $\mathcal{T}_2$ is a deductive $\Sigma$-conservative extension of $\mathcal{T}_1$ w.r.t. $\mathcal{L}$ for $\Sigma = sig(\mathcal{T}_1)$.*

In other words, an ontology $\mathcal{T}_2$ is a deductive $\Sigma$-conservative extension of $\mathcal{T}_1$ for a signature $\Sigma$ and language $\mathcal{L}$ iff every logical consequence $\alpha$ of $\mathcal{T}_1$ constructed using the language $\mathcal{L}$ and names only from $\Sigma$, is already a logical consequence of $\mathcal{T}_2$; that is, the additional axioms in $\mathcal{T}_2 \backslash \mathcal{T}_1$ do not result into new logical consequences over the vocabulary $\Sigma$. Note that if $\mathcal{T}_2$ is a deductive $\Sigma$-conservative extension of $\mathcal{T}_1$ w.r.t. $\mathcal{L}$, then $\mathcal{T}_2$ is a deductive $\Sigma'$-conservative extension of $\mathcal{T}_1$ w.r.t. $\mathcal{L}$ for every $\Sigma' \subseteq \Sigma$. The notion of a deductive conservative extension can be directly applied to our ontology reuse scenario.

**Definition 2 (Safety for a TBox)** *Given $\mathcal{L}$-TBoxes $\mathcal{T}$ and $\mathcal{T}'$, we say that $\mathcal{T}$ is* safe *for $\mathcal{T}'$ (or $\mathcal{T}$ imports $\mathcal{T}'$ in a safe way) w.r.t. $\mathcal{L}$ if $\mathcal{T} \cup \mathcal{T}'$ is a deductive conservative extension of $\mathcal{T}'$ w.r.t. $\mathcal{L}$.*

Hence, the first reasoning task relevant to our ontology reuse scenario can be formulated as follows: given $\mathcal{L}$-TBoxes $\mathcal{T}$ and $\mathcal{T}'$, determine if $\mathcal{T}$ is safe for $\mathcal{T}'$ w.r.t. $\mathcal{L}$.

We have shown that, given $\mathcal{P}$ consisting of axioms P1—P4, E1, E2, and $\mathcal{Q}$ consisting of axioms M1—M5 from Figure 1, there exists an axiom $\delta := \text{Cystic\_Fibrosis} \sqsubseteq \bot$ that uses only names in $\text{sig}(\mathcal{Q})$ such that $\mathcal{Q} \not\models \delta$ but $\mathcal{P} \cup \mathcal{Q} \models \delta$. According to Definition 1, this means that $\mathcal{P} \cup \mathcal{Q}$ is not a deductive conservative extension of $\mathcal{Q}$ w.r.t. any language $\mathcal{L}$ in which $\delta$ can be expressed (e.g. $\mathcal{L} = \mathcal{ALC}$). It is possible, however, to show that if the axiom E2 is removed from $\mathcal{P}$ then for the resulting ontology $\mathcal{P}_1 = \mathcal{P} \backslash \{\text{E2}\}$, $\mathcal{P}_1 \cup \mathcal{Q}$ is a deductive conservative extension of $\mathcal{Q}$. The following notion is useful for proving deductive conservative extensions:

**Definition 3 (Model Conservative Extension)** *Let $\mathcal{T}_1$ and $\mathcal{T}_1 \subseteq \mathcal{T}_2$ be two $\mathcal{L}$-TBoxes and $\Sigma$ a signature over the description logic $\mathcal{L}$. We say that $\mathcal{T}_2$ is a* model $\Sigma$-conservative extension *of $\mathcal{T}_1$, if for every model $\mathcal{I}_1$ of $\mathcal{T}_1$, there exists a model $\mathcal{I}_2$ of $\mathcal{T}_2$ such that $\mathcal{I}_1|_\Sigma = \mathcal{I}_2|_\Sigma$, i.e., the extensions of concept and role names from $\Sigma$ coincide in $\mathcal{I}_1$ and $\mathcal{I}_2$. We say that $\mathcal{T}_2$ is a model conservative extension of $\mathcal{T}_1$ if $\mathcal{T}_2$ is a model $\Sigma$-conservative extension of $\mathcal{T}_1$ for $\Sigma = sig(\mathcal{T}_1)$.*

Observe that the definition of conservative extension provided in the lectures (Definition 6.3) does not impose the requirement that $\mathcal{T}_1$ must be a subset of $\mathcal{T}_2$. Therefore, Definition 6.3 can be considered a generalization of the definition provided in Definition 3 above. The conditions (1) and (2) in Definition 6.3 are immediate consequences of the fact that $\mathcal{T}_1$ is a subset of $\mathcal{T}_2$. Definition 3 specifically states that the axioms in $\mathcal{T}_2 \backslash \mathcal{T}_1$ do not affect the semantics of the names in $\Sigma$. It is important to note that $\mathcal{T}_1$ can include additional names beyond those in $\Sigma$, and their semantics may be subject to change.

The notion of model conservative extension in Definition 3 can be seen as a semantic counterpart for the notion of deductive conservative extension in Definition 1: the latter is defined in terms of "logical consequences", whereas the former is defined in terms of "models". Intuitively, a TBox $\mathcal{T}_2$ is a model $\Sigma$-conservative

extension of $\mathcal{T}_1$ if for every model of $\mathcal{T}_1$ one can find a model of $\mathcal{T}_2$ over the same domain which interprets the names from $\Sigma$ in the same way. The notion of model conservative extension, however, does not provide a complete characterization of deductive conservative extensions, as given in Definition 1; that is, this notion can be used for proving that an ontology is a deductive conservative extension of another, but not vice versa.

- Show that for every two $\mathcal{L}$-TBoxes $\mathcal{T}_2$, $\mathcal{T}_1 \subseteq \mathcal{T}_2$, and a signature $\Sigma$ over $\mathcal{L}$, if $\mathcal{T}_2$ is a model $\Sigma$-conservative extension of $\mathcal{T}_1$, then it is a also deductive $\Sigma$-conservative extension of $\mathcal{T}_1$.

- Show that there exists two $\mathcal{ALC}$-TBoxes $\mathcal{T}_2$, $\mathcal{T}_1 \subseteq \mathcal{T}_2$ such that $\mathcal{T}_2$ is a deductive $\Sigma$-conservative extension of $\mathcal{T}_1$ but it is NOT a model conservative extension of $\mathcal{T}_1$.

- Consider the TBox $\mathcal{P}_1$ consisting of the axioms P1—P4 and E1 and the TBox $\mathcal{Q}$ consisting of the axioms M1—M5 from Figure 1. Show that $\mathcal{P}_1 \cup \mathcal{Q}$ is a deductive conservative extension of $\mathcal{Q}$.