Assignment#3

Knowledge Representation and Processing

June 1, 2021

Note: Assignment#3, due on 14:00 June 1, contributes to 10% of the total mark of the course.

Q1. A satisfiable class must always have a non-empty interpretation.

(A) True (B) False (C) Don't know

Answer:A

Q2. An unsatisfiable class may have a non-empty interpretation in some model.

(A) True (B) False (C) Don't know

Answer:B

Q3. An unsatisfiable class will be a subclass of any other class.

(A) True (B) False (C) Don't know

Answer:A

Q4. Given the following ontology definition.

- Class: PetOwner
- SubClassOf: Person that hasPet some Cat and hasPet only (Cat or Dog)
- Class: Cat
- SubClassOf: Animal
- Class: Dog
- SubClassOf: Animal
- Individual: Fred
- Types: PetOwner

- Facts: hasPet Tibbs
- Individual: Tibbs
- Types: not Dog
- *What additional information do we know about Tibbs?*

Answer:Tibbs is a cat.

Q5. Given the following ontology definition.

- Class: PetOwner
- SubClassOf: Person that hasPet some Cat and hasPet some Dog
- Class: Cat
- SubClassOf: Animal
- Class: Dog
- SubClassOf: Animal
- Individual: Fred
- Types: PetOwner
- Facts: hasPet Tibbs hasPet Fido
- Individual: Tibbs
- Types: Cat
- Individual: Fido
- *Is Fido a Dog?*

Answer:Yes.

Q6. Given the following ontology definition.

- Class: PetOwner
- SubClassOf: Person that hasPet some Cat
- *Is owning a Cat enough to recognize a Person as a PetOwner?*

Answer:No.

Q7. Given the following ontology definition.

- Class: PetOwner

- SubClassOf: Person that hasPet some Cat and hasPet only Cat

- *How many Cats does a PetOwner have as Pets?*

Answer:I don't know, but at least one.

Q8. What are Competency Questions? How would you use them during ontology development? Answer using around 150 words.
Answer:

Competency questions are natural language questions outlining and constraining the scope of knowledge represented in an ontology. They play a significant role in ontology testing and represent functional requirements in the sense that the developed ontology should be able to answer them. It mainly ensures ontology quality in two ways, the first one is that it enables developers to identify the elements and the relationships between them. The other is that it allows the developers to check the requirements satisfiability in a much more simple way.

As for how to use it, there are two time frames, the first is requirements specification and the second is evaluation phase. The developers just need to create some competency questions and give the expected answers. Then we execute the test and check whether it passes.

Additionally, I've read some papers related to CQchecker(a tool to support the automation of verifying CQs against ontologies), maybe it can be used in our ontology development.

Q9. In Assignment#1, you produced a basic hierarchy of sushi ingredients. For this assignment, you are to repeat the exercise of producing an ontology (based on the one you produced for Assignment#1), but using the experience and information that you have gained along with the Competency Question you have been identifying. Your ontology should be sufficient to enable the set of Competency Questions for the ontology that you, as a class, will formulate. These Competency Questions are those that you would wish an intelligent, ontology driven restaurant sushi menu to be able to answer. These Competency Questions will form the basis for evaluating your ontology. There are several things you will do in developing your sushi ontology:

- Produce some competency questions that will guide the ontology's content, design and evaluation what should the ontology be "competent" to answer"?

- Produce a hierarchy of sushi ingredients such that each type of sushi can be described (This was supposed to be done in Assignment#1).

- For each class, identify adjectives (in the menu) that modify the ingredient and include them as *comments*.

- Design the axioms patterns that allow sushi and platters of sushi to be described according to the competency questions.

- Produce the sushi ontology, that covers both sushi and platters of sushi, that will allow a broad range of queries from the competency to be answered.

  Submit an OWL ontology sushi-XXX.owl, for XXX your student id.

Answer:
Questions my sushi ontology can answer:
(1) Are there any salmon maki rolls?
(2) Does this sushi have salmon?
(3) Find all sushi that have crumbs.
(4) How many salmon maki rolls are there in the mined maki plate?
(5) Find all the grain that can be used as ingredient.
(6) If I were a vegetarian, which one can I purchase?
(7) Find all the sushi I can choose from if I love meat.
(8) Are there any fruits other than avocado as ingredients for sushi?

Q10. If you had only existential restriction ($\exists$) or only universal restriction ($\forall$) in your ontology, which, from a modelling perspective, would be preferable, and why? (approx. 150 words)
Answer:

For me, I would prefer the existential restriction, but one significant fact is that each of them has the same power to express what we need. (Because $\exists r.A$ is equal to $\forall r.(\neg A)$.)

So why I prefer the existential restriction? Two main reasons are as follows. First of all, when giving an interpretation manually, $(\exists r.A)^{\mathcal{I}}$ is much more easier for me to caculate than $(\forall r.A)^{\mathcal{I}}$. To compute $(\exists r.A)^{\mathcal{I}}$, I just need to traverse $r^{\mathcal{I}}$ and then select the first element whose second element appears in $A^{\mathcal{I}}$. But when caculating $(\forall r.A)^{\mathcal{I}}$, I have to traverse all x in $\Delta^{\mathcal{I}}$ and every y in $\Delta^{\mathcal{I}}$ for one x whether there is a relation between x and y. To be honest, this process is painful. The other reason is that computing $(\exists r.A)^{\mathcal{I}}$ is based on $A^{\mathcal{I}}$ and $r^{\mathcal{I}}$, but $(\forall r.A)^{\mathcal{I}}$ needs to start from $\Delta^{\mathcal{I}}$, which means when $A^{\mathcal{I}}$ and $r^{\mathcal{I}}$ is small but $\Delta^{\mathcal{I}}$ is huge, the process of computing $(\exists r.A)^{\mathcal{I}}$ will be finished faster. (Of course, even if $A^{\mathcal{I}}$ is equal to $\Delta^{\mathcal{I}}$, $(\exists r.A)^{\mathcal{I}}$ is still faster.)

Therefore, I prefer the existential restriction.

Q11. Apply the $\mathcal{ALC}$-tableaux algorithm to the following concepts and determine which are satisfiable and which are not. If a concept is satisfiable, given an interpretation satisfying it.

- $A \sqcap \neg A$
- $\exists r.\exists r.(A \sqcap \neg A)$
- $\forall r.\forall r.(A \sqcap \neg A)$
- $\exists r.A \sqcap \forall s.\neg A$

- $\exists r.A \sqcap (\forall r.\neg A \sqcap \exists r.\neg A)$

Answer:
$A \sqcap \neg A$:No
$S_0 = \{x : A \sqcap \neg A\}$
$\rightarrow_{\sqcap}$: $S_1 = S_0 \cup \{x : A, x : \neg A\}$
we have obtained a clash, so $A \sqcap \neg A$ is not satisfiable

$\exists r.\exists r.(A \sqcap \neg A)$:No
$S_0 = \{x : \exists r.\exists r.(A \sqcap \neg A)\}$
$\rightarrow_{\exists}$: $S_1 = S_0 \cup \{(x, y) : r, y : \exists r.(A \sqcap \neg A)\}$
$\rightarrow_{\exists}$: $S_2 = S_1 \cup \{(y, z) : r, z : A \sqcap \neg A\}$
$\rightarrow_{\sqcap}$: $S_3 = S_2 \cup \{z : A, z : \neg A\}$
we have obtained a clash, so $\exists r.\exists r.(A \sqcap \neg A)$ is not satisfiable

$\forall r.\forall r.(A \sqcap \neg A)$:Yes
Interpretation:
$\Delta^{\mathcal{I}} = \{a, b, c, d\}, A^{\mathcal{I}} = \{b, d\}, r^{\mathcal{I}} = \{(a, b), (a, c)\}$
$(\forall r.(A \sqcap \neg A))^{\mathcal{I}} = \{b, c, d\}$
$(\forall r.\{b, c, d\})^{\mathcal{I}} = \{a, b, c, d\}$
so it is satisfiable

$\exists r.A \sqcap \forall s.\neg A$:Yes
Interpretation:
$\Delta^{\mathcal{I}} = \{a, b, c, d\}, A^{\mathcal{I}} = \{c, d\}, r^{\mathcal{I}} = \{(a, c), (b, d)\}, s^{\mathcal{I}} = \{(a, a)\}$
$(\exists r.A)^{\mathcal{I}} = \{a, b\}, (\forall s.\neg A)^{\mathcal{I}} = \{a, b, c, d\}$
$(\exists r.A \sqcap \forall s.\neg A)^{\mathcal{I}} = \{a, b\}$
so it is satisfiable

$\exists r.A \sqcap (\forall r.\neg A \sqcap \exists r.\neg A)$:No
$S_0 = \{x : \exists r.A \sqcap (\forall r.\neg A \sqcap \exists r.\neg A)\}$
$\rightarrow_{\sqcap}$: $S_1 = S_0 \cup \{x : \exists r.A, x : \forall r.\neg A \sqcap \exists r.\neg A\}$
$\rightarrow_{\sqcap}$: $S_2 = S_1 \cup \{x : \forall r.\neg A, x : \exists r.\neg A\}$
$\rightarrow_{\exists}$ $(for \exists r.A) : S_3 = S_2 \cup \{(x, y) : r, y : A\}$
$\rightarrow_{\forall}$: $S_4 = S_3 \cup \{y : \neg A\}$
we have obtained a clash($\{y : A, y : \neg A\}$), so $\exists r.A \sqcap (\forall r.\neg A \sqcap \exists r.\neg A)$ is not satisfiable

Q12. Use the $\mathcal{ALC}$-tableaux algorithm to determine whether $\emptyset \models \forall r.A \sqsubseteq \exists r.A$.
Answer:
$\emptyset \models \forall r.A \sqsubseteq \exists r.A \Leftrightarrow \forall r.A \sqcap \neg\exists r.A\ is\ not\ satisfiable\ w.r.t\ \emptyset$
$First\ turn\ it\ into\ NNF \rightarrow \forall r.A \sqcap \forall r.\neg A$
$S_0 = \{x : \forall r.A \sqcap \forall r.\neg A\}$
$\rightarrow_{\sqcap}$: $S_1 = S_0 \cup \{x : \forall r.A, x : \forall r.\neg A\}$
now no rules can be used
and $\forall r.A \sqcap \neg\exists r.A\ is\ satisfiable$

so $\emptyset \models \forall r.A \sqsubseteq \exists r.A$ is wrong

Q13. Let $\mathcal{T}$ be a general $\mathcal{EL}$ TBox, C, D $\mathcal{EL}$ concepts and A, B concept names not occurring in $\mathcal{T}$ or C, D. *Show that:*

$$\mathcal{T} \models C \sqsubseteq D \textit{ if and only if } \mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\} \models A \sqsubseteq B.$$

Answer:
$\mathcal{T} \models C \sqsubseteq D\ means\ \forall \mathcal{I} \models \mathcal{T}, \mathcal{I} \models C \sqsubseteq D$
$let\ \mathcal{T}' = \mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\}$
$\Rightarrow$:
$\forall \mathcal{I}' \models \mathcal{T}', we\ have\ \mathcal{I}' \models \mathcal{T}\ and\ \mathcal{I}' \models \{A \sqsubseteq C, D \sqsubseteq B\}$
$\mathcal{I}' \models \mathcal{T}\ means\ \mathcal{I}' \models C \sqsubseteq D \to C^{\mathcal{I}'} \subseteq D^{\mathcal{I}'}$
$\mathcal{I}' \models \{A \sqsubseteq C, D \sqsubseteq B\}\ means\ A^{\mathcal{I}'} \subseteq C^{\mathcal{I}'}, D^{\mathcal{I}'} \subseteq B^{\mathcal{I}'} \to A^{\mathcal{I}'} \subseteq B^{\mathcal{I}'}$
$now\ we\ have\ \forall \mathcal{I}' \models \mathcal{T}',\ \mathcal{I}' \models A \sqsubseteq B, i.e. \mathcal{T}' = \mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\} \models A \sqsubseteq B$

$\Leftarrow$:
we have $\mathcal{T}' = \mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\} \models A \sqsubseteq B$
let's suppose $\mathcal{T} \not\models C \sqsubseteq D$
so $\exists \mathcal{I}_0 \models \mathcal{T}$ and $\mathcal{I}_0 \not\models C \sqsubseteq D$
expand $\mathcal{I}_0$, let $A^{\mathcal{I}_0} = C^{\mathcal{I}_0}$,$B^{\mathcal{I}_0} = D^{\mathcal{I}_0}$, we call it $\mathcal{I}'_0$
it's obvious $\mathcal{I}'_0 \models \{A \sqsubseteq C, D \sqsubseteq B\}$
because A,B not occurring in $\mathcal{T}$ or C,D, so $\mathcal{I}'_0 \models \mathcal{T}$
so $\mathcal{I}'_0 \models \mathcal{T}'$
but $\mathcal{I}_0 \not\models C \sqsubseteq D \to C^{\mathcal{I}_0} \not\subseteq D^{\mathcal{I}_0} \to\ A^{\mathcal{I}'_0} \not\subseteq B^{\mathcal{I}'_0}$
which means $\mathcal{I}'_0 \models \mathcal{T}'$ but $\mathcal{I}'_0 \not\models A \sqsubseteq B$
this is contradict with $\mathcal{T}' \models A \sqsubseteq B$ so our suppose is not true, *i.e.* $\mathcal{T} \models C \sqsubseteq D$
To sum up, we've proved
$\mathcal{T} \models C \sqsubseteq D$ if and only if $\mathcal{T} \cup \{A \sqsubseteq C, D \sqsubseteq B\} \models A \sqsubseteq B$

Q14. Recall that a general $\mathcal{EL}$ TBox $\mathcal{T}$ is *in normal form* (*normalized*) if it only contains GCIs of the following form:

$$A \sqsubseteq B \qquad A_1 \sqcap A_2 \sqsubseteq B \qquad A \sqsubseteq \exists r.B \qquad \exists r.A \sqsubseteq B$$

where A, A_1, A_2, B are concept names or the top concept $\top$, and r is a role name. One can normalize a given $\mathcal{EL}$ TBox by exhaustively applying the following normalization rules (slightly different from those present in the lecture slides).

- $M \sqsubseteq N \longrightarrow M \sqsubseteq A, A \sqsubseteq N$
- $C \sqcap M \sqsubseteq B \longrightarrow M \sqsubseteq A, C \sqcap A \sqsubseteq B$
- $M \sqcap C \sqsubseteq B \longrightarrow M \sqsubseteq A, A \sqcap C \sqsubseteq B$
- $\exists r.M \sqsubseteq B \longrightarrow M \sqsubseteq A, \exists r.A \sqsubseteq B$
- $B \sqsubseteq \exists r.M \longrightarrow A \sqsubseteq M, B \sqsubseteq \exists r.A$
- $B \sqsubseteq D \sqcap E \longrightarrow B \sqsubseteq D, B \sqsubseteq E$

- $C \equiv D \longrightarrow C \sqsubseteq D, D \sqsubseteq C$

where C, D, E are arbitrary $\mathcal{EL}$ concepts, M, N are $\mathcal{EL}$ concepts that are neither concept names nor $\top$, B is a concept name, and A is a fresh concept name.
Show that: Any general $\mathcal{EL}$ TBox $\mathcal{T}$ can be transformed into a normalized one $\mathcal{T}'$ by a linear number of applications of the normalization rules.
Answer:
suppose there are n axioms in $\mathcal{EL}$ TBox
consider the axiom with the largest number of concept names(we express the number with m) appear in it

First of all, if the axiom contains $\equiv$ then use the rule to turn it into two ... $\sqsubseteq$ ...

here we only dicuss the axiom that already in the form like $M \sqsubseteq N$

Let $A[1,k]$ stands for M which has k concept names appeared,$A[k+1,m]$ stands for N which has $m-k$ concept names appeared

To begin with, we use 1 rule and get $A[1,k] \sqsubseteq B, B \sqsubseteq A[k+1,m]$,then we normalize $A[1,k] \sqsubseteq B$. now we have two situations:(1)$A[1,t]$ and $A[t+1,k]$ is linked by $\sqcap$;(2)there is a $\exists r.$ in front of $A[1,k]$, in case(1):we use rule and get $A[1,t] \sqsubseteq C, A[t+1,k] \sqsubseteq D, C \sqcap D \sqsubseteq B$, in case(2):we use a rule and get $A[1,k] \sqsubseteq C, \exists r.C \sqsubseteq B$ because $\exists$ don't the length of a expression, so we let the number of $\exists$ which need to be processed is α. Keep repeating the above process, we can normalize $A[1,k] \sqsubseteq B$ with at most $\alpha + \sum_{i=0}^{\frac{k-1}{2}} 2^i$ times using the rule. Processing $B \sqsubseteq A[k+1,m]$ is in the same way.

So processing the largest axiom we need use at most $R = 2 * (\alpha + \sum_{i=0}^{\frac{k-1}{2}} 2^i)$ times of rules,which is a constant in fact. Then normalizing the $\mathcal{EL}$ *TBox* we need use $n \cdot R$ times, which is a linear number.

Q15. Translate the following statements to concept inclusions in the description logic $\mathcal{SHOIQ}$. State which symbols are used as concept names, role names and nominals.

- Every student at Nanjing University is a human being.
- Nanjing University has at least 30,000 students.
- Every citizen of China is an Asia.
- East Asia consists of at least 5 countries.
- The domain of the relation "citizen of" consists of human beings.
- The range of the relation "citizen of" consists of countries.
- There are at least 1,000,000,000 Chinese citizens.

  Ensure that it follows from your translation that China is a country.

Answer:
concept names: *Asia*, *Country*, *Human*

role names: $consist\_of$ $citizen\_of$, $student\_in$
nominals: $\{China\}$, $\{East\_Asia\}$, $\{NJU\}$

- $\exists \mathsf{student\_in}.\{\mathsf{NJU}\} \sqsubseteq \mathsf{Human}$
- $\{\mathsf{NJU}\} \sqsubseteq (\geq 30000\, \mathsf{student\_at}^{-}.\top)$
- $\exists \mathsf{citizen\_of}.\{\mathsf{China}\} \sqsubseteq \mathsf{Asia}$
- $\{\mathsf{East\_Asia}\} \sqsubseteq (\geq 5\, \mathsf{consists\_of.country})$
- $\exists \mathsf{citizen\_of}.\top \sqsubseteq \mathsf{Human}$
- $\exists \mathsf{citizen\_of}^{-}.\top \sqsubseteq \mathsf{Country}$
- $\{\mathsf{China}\} \sqsubseteq (\geq 1000000000\, \mathsf{citizen\_of}^{-}.\top)$

In the last concept inclusion the inverse of citizen_of ensures that China is a country.