Semantic Web: Tutorial

October 23, 2016

Exercise n^o 1

Given the following facts:

- If Pierre comes, we play cards;
- If Pierre and Jean come, there will be disputes.
- If we don't play cards, there won't be any dispute.
- Pierre does not come.

Represent the four pieces of information/knowledge given above in a logical langage and ask the following question: can we deduce from the knowledge that there won't be dispute?

Answer: We can use propositional logic to solve the question.

Let's take P for " Pierre comes", C for "we play cards", J for "Jean comes", and D for " there will be disputes". So we get the following knowledge base: $K = \{P \rightarrow C, P \wedge J \rightarrow D, \neg C \rightarrow \neg D, \neg P\}$. The question "can we deduce from the knowledge that there won't be dispute" is to answer: does it hold $K \vdash \neg D$? There are two ways to check this (you can just use one of them): one via semantics (truth table), and the other is with syntax (for example: resolution which is not introduced in our course so we omit it here).

- (Semantic way) By definition of semantics of propositional logic, $K \vdash \alpha$ if and only if each interpretation satisfying all the formulae in K also satisfies α. We consider all possible interpretations:

P	C	J	D	$P \rightarrow C$	$P \wedge J \rightarrow D$	$\neg C \rightarrow \neg D$	$\neg P$	$\neg D$
t	t	t	t	...				
t	t	t	f	...				
t	t	f	t	...				
t	f	t	t	...				
f	t	t	t	...				
t	t	f	f	...				
t	f	f	t	...				
f	f	t	t	...				
t	f	t	f	...				
f	t	t	f	...				
f	t	f	t	...				
f	f	f	t	...				
f	f	t	f	...				
f	t	f	f	...				
t	f	f	f	...				
f	f	f	f	...				

However, we can have an interpretation I with $P^I = false, C^I = D^I = true, J^I = true$ that satisfies each formula in K, but does not satisfy $\neg D$ (because $D^I = true$). So we cannot deduce $\neg D$ from the four facts.

Exercise n^o 1bis

Given the following facts:

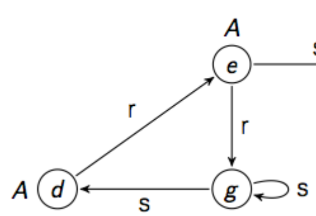
- If Pierre comes, we play cards;
- If Pierre and Jean come, there will be disputes.
- If there are disputes, it means we play cards.
- Pierre comes.

Which logical langage can be used to represent these four pieces of information/knowledge? Can we deduce from the knowledge that there will be disputes?

Hints: We can use Horn Rule (which is a sub-language of propositional logic) to solve the question. Then you can either use truth table as the previous exercise to answer the reasoning task or use forward or backward chaining algorithm (which is the syntactic approach for Horn Rule). Please revise these algorithms yourselves.

Exercise n^o 2

Consider the interpretation (the graphic representation) I with $\Delta^I = \{d, e, f, g\}$:



For each concept C of the description logic $\mathcal{ALC}$, enumerate all the elements x of Δ^I such that $x \in C^I$:

- A, B, $\neg A \sqcap B$

  Answer: the instances of concept A under this interpretation is $\{e, f, d\}$; The instances of concept B under this interpretation is $\{f\}$; The instances of concept $\neg A \sqcap B$ are the intersection of those of $\neg A$ and those of B. For $\neg A$, they are $\{g\}$; For B, they are $\{f\}$. So the intersection is empty, that is, no member of Δ^I belongs to the concept $\neg A \sqcap B$ by this interpretation.

- $\exists s.A$, $\forall s.A$

  Answer: The semantics of $\exists s.A$ is defined as follows:

  $$(\exists s.A)^I = \{x \in \Delta^I \mid \text{there exists } y \in \Delta^I \text{ such that } s(x, y) \text{ and } y \in \Delta^I\}.$$

  Therefore, $(\exists s.A)^I = \{e, g\}$ because for e, there is f satisfying $s(e, f)$ and $f \in A^I$. Similarly, $g \in (\exists s.A)^I$ due to d.
  The semantics of $\forall s.A$ is defined as follows:

  $$(\forall s.A)^I = \{x \in \Delta^I \mid \text{for any } y \in \Delta^I, \text{ if } s(x, y), \text{ then } y \in \Delta^I\}.$$

  Therefore, $(\forall s.A)^I = \{e\}$ because for e, any element having a relation with e is of type A. But g is not any more in $(\forall s.A)^I$.

- $\exists s.\exists s.\exists s.\exists s.A$

  Answer: by successively applying the definition of $\exists s.A$, we get $(\exists s.\exists s.\exists s.\exists s.A)^I = \{g\}$.

Exercise n^o 3

Is the ontology $O = (TBox, ABox)$ where $TBox = \{A \sqsubseteq B \sqcup C\}$ and $ABox = \{A(a), \neg B(a)\}$ satisfiable? If yes, give an interpretation satisfying O; Otherwise, justify your answer. The same question for $O' = O \cup \{\neg C(a)\}$.

Answer: An ontology is satisfiable if and only if we can find an interpretation (something like the graph in the previous exercise) that satisfies each formula in its TBox and ABox. For O, we can define an interpretation I with domain $\Delta^I = \{a, b\}$, and $A^I = \{a, b\}, B^I = \{b\}, C^I = \{a\}$. We can see that I satisfies $A \sqsubseteq B \sqcup C$ and $\{A(a), \neg B(a)\}$. So O is satisfiable.

However, for O', we cannot find such an interpretation because by the definition of semantics for $A \sqsubseteq B \sqcup C$, for any interpretation I of this axiom, this holds that $A^I \subseteq B^I \cup C^I$. In contrast, an interoperation I satisfies n this case, it is impossible for I to satisfy furthermore that $\{A(a), \neg B(a), \neg C(a)\}$ requires that a is interpretated as a member of A but neither a member of B nor C, that is, $a^I \in A^I, a^I \notin B^I, a^I \notin C^I$, which contradicts with the requirement $A^I \subseteq B^I \cup C^I$. So O' is unsatisfiable.

Exercise n^o 4 (Mad cows cannot exist)

Now suppose $\mathcal{T}$ contains the following axioms:

$$
\begin{aligned}
Sheep &\sqsubseteq Animal \sqcap \forall Eats.Grass && (1)\\
Cow &\sqsubseteq Vegetarian && (2)\\
MadCow &= Cow \sqcap \exists Eats.(Brain \sqcap \exists PartOf.Sheep) && (3)\\
Vegetarian &= Animal \sqcap (\forall Eats.\neg Animal) \sqcap (\forall Eats.\neg(\exists PartOf.Animal)) && (4)\\
Animal \sqcup \exists PartOf.Animal \sqsubseteq &\sqsubseteq \neg(Plant \sqcup \exists PartOf.Plant) && (5)
\end{aligned}
$$

Prove that MadCow is unsatisfiable w.r.t. $\mathcal{T}$.

Proof: By the third inclusion, we have

$$\mathcal{T} \models MadCow \sqsubseteq Cow.$$

We also know that $\mathcal{T} \models Cow \sqsubseteq Vegetarian$, so

$$\mathcal{T} \models MadCow \sqsubseteq Vegetarian.$$

Because using the definition of Vegetarian (fourth inclusion), we get

$$\mathcal{T} \models MadCow \sqsubseteq \forall Eats.\neg(\exists PartOf.Animal).$$

By combining the first and third inclusions, we obtain

$$\mathcal{T} \models MadCow \sqsubseteq \exists Eats.(\exists PartOf.Animal).$$

Putting this together, we have

$$\mathcal{T} \models MadCow \sqsubseteq \exists Eats.(\exists PartOf.Animal) \sqcap \forall Eats.\neg(\exists PartOf.Animal)$$

Since the latter concept is always empty (why?), it follows that $MadCow^I = \emptyset$ in every model I of $\mathcal{T}$.

Exercice n^o 5

Consider the following $\mathcal{ALC}$ ontology O:

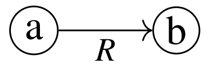
$$
\begin{aligned}
\mathcal{T} &= \{A \sqsubseteq \forall R.B, B \sqsubseteq \neg F, E \sqsubseteq G, A \sqsubseteq D \sqcup \neg E, D \sqsubseteq \exists R.F, \exists R.\neg B \sqsubseteq G\}\\
\mathcal{A} &= \{A(a), R(a, b), F(b)\}
\end{aligned}
$$

1. Is $O = (\mathcal{T}, \mathcal{A})$ satisfiable?

   Answer: Suppose that I is a model of O. Then we must have: $a^I \in A^I, (a^I, b^I) \in R^I, b^I \in F^I$. Since $A \sqsubseteq \forall R.B \in \mathcal{T}, a^I \in A^I$,and $(a^I, b^I) \in R^I$,we get $b^I \in B^I$. Because of the inclusion $B \sqsubseteq \neg F$, we must also have $b^I \in (\neg F)^I$. This is a contradiction: b^I cannot belong both to F^I and $(\neg F)^I$!!! As we have reached a contradiction, our original assumption that O is satisfiable must be wrong.

2. Is the concept $A \sqcap \exists R.\neg B$ satisfiable w.r.t. an empty TBox (or $A \sqcap \exists R.\neg B$ satisfiable for short)?

   Answer (proof by semantics): by definition, if we can find an interpretation I such that $(A \sqcap \exists R.\neg B)^I \neq \emptyset$, then $A \sqcap \exists R.\neg B$ satisfiable. Yes, we can find one as follows:

   Answer (proof by syntax – Tableau algorithm):
   We start with $\mathcal{S} = \{\mathcal{A}_0\}$ where $\mathcal{A}_0 = A \sqcap \exists R.\neg B(a_0)$.
   Apply $\sqcap$-rule to $\mathcal{A}_0$:
   get $\mathcal{S}' = \{\mathcal{A}_0'\}$ where $\mathcal{A}_0' = \mathcal{A}_0 \cup \{A(a_0), \exists R.\neg B(a_0)\}$.
   Apply $\exists$-rule to $\mathcal{A}_0'$:
   get $\mathcal{S}'' = \{\mathcal{A}_0''\}$ where $\mathcal{A}_0'' = \mathcal{A}_0' \cup \{R(a_0, a_1), \neg B(a_1)\}$.
   $\mathcal{A}_0''$ is complete and contains no clash, so the original concept is satisfiable.

3. Is the concept $A \sqcap \exists R.\neg B$ satisfiable w.r.t. $\mathcal{T}$?

   Answer (proof by semantics): suppose $A \sqcap \exists R.\neg B$ satisfiable w.r.t. $\mathcal{T}$, which means there is a model of $\mathcal{T}$ such that $(A \sqcap \exists R.\neg B)^I \neq \emptyset$. Assume $a \in \Delta^I$ and $a \in (A \sqcap \exists R.\neg B)^I$. That is, $a \in A^I$ and $a \in (\exists R.\neg B)^I$. Since $a \in (\exists R.\neg B)^I$, there is a $b \in \Delta^I$ such that $(a, b) \in R^I$ and $b \in (\neg B)^I$. By $a \in A^I$ and $A \sqsubseteq \forall R.B \in \mathcal{T}$, so $a \in (\forall R.B)^I$. Due to the fact that $(a, b) \in R^I$, we deduce that $b \in B^I$, which contradicts with $b \in (\neg B)^I$. Therefore, the concept is unsatisfiable w.r.t. $\mathcal{T}$.

   Answer (proof by syntax): By tableau algorithm:
   We start with $\mathcal{S} = \{\mathcal{A}_0\}$ where $\mathcal{A}_0 = A \sqcap \exists R.\neg B(a_0)$.
   Apply $\sqcap$-rule to $\mathcal{A}_0$:
   get $\mathcal{S}' = \{\mathcal{A}_0'\}$ where $\mathcal{A}_0' = \mathcal{A}_0 \cup \{A(a_0), \exists R.\neg B(a_0)\}$.
   Apply $\exists$-rule to $\mathcal{A}_0'$:
   get $\mathcal{S}'' = \{\mathcal{A}_0''\}$ where $\mathcal{A}_0'' = \mathcal{A}_0' \cup \{R(a_0, a_1), \neg B(a_1)\}$.
   Apply $\sqsubseteq^{at}$-rule to $\mathcal{A}_0''$:
   (since a_0, a_1 are not blocked, and $A \sqsubseteq \forall R.B \in \mathcal{T}$ but $\forall R.B(a) \notin \mathcal{A}_0''$ for $a \in \{a_0, a_1\}$)
   get $\mathcal{S}''' = \{\mathcal{A}_0'''\}$ where $\mathcal{A}_0''' = \mathcal{A}_0'' \cup \{B(a_1)\}$.
   all the ABox in $\mathcal{S}'''$ (i.e. $\mathcal{A}_0'''$) contains a clash $\{\neg B(a_1), B(a_1)\}$, so the original concept is unsatisfiable.

Exercise n^o 6

Check the satisfiability of the following concept (without TBox):

$$(A \sqcup \exists R.B) \sqcap (\neg A \sqcap \forall R.\neg B).$$

If it is satisfiable, give the interpretation corresponding to your tableau construction.
Answer: We can use the figure representation for the tableaux construction procedure, as shown in Figure 1: And the interpretation $I = (\Delta^I, \cdot^I)$ induced by the branch $B2$ is:

$$\Delta^I = \{a_0\}, \quad A^I = \{a_0\}, \quad B^I = R^I = \emptyset.$$

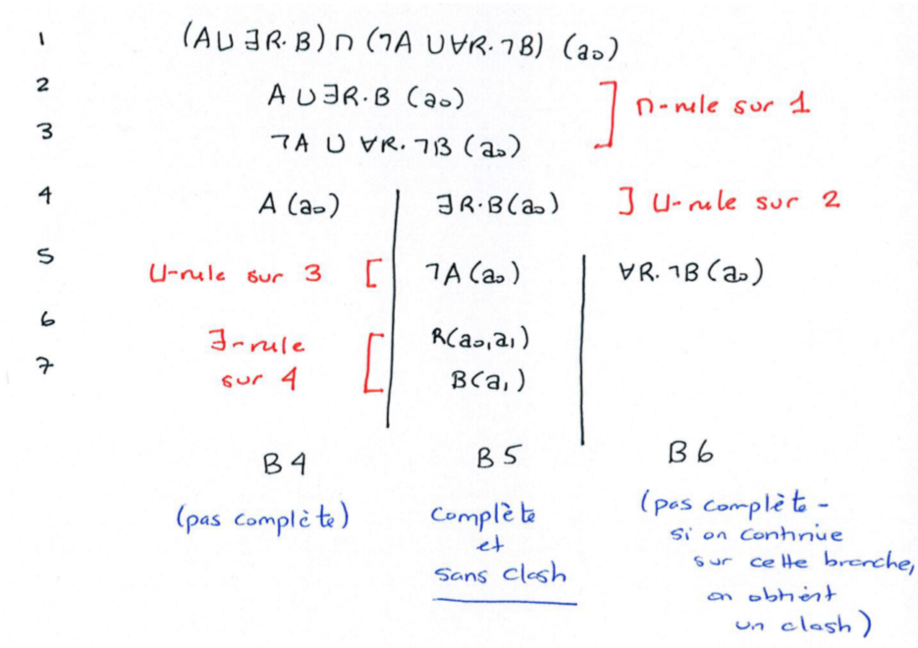


Figure 1: The first tableau construction

There are other possible construction due to different orders of the rules applied, as shown in , as shown in Figure 2: And the interpretation $I = (\Delta^I, \cdot^I)$ induced by the branch $B5$ is:

$$\Delta^I = \{a_0, a_1\}, \quad A^I = \emptyset, \quad B^I = \{a_1\}, R^I = \{(a_0, a_1)\}.$$

NOTE:

- For this concept, we don't need to finish all the branches because the Tableau algorithm stops when there is a complete branch without clash.
- The interpretation asked is the one corresponding to the complete brach without clash, not any arbitrary one.

Exercise n^o 7

Prove the following equivalences:

$$\begin{array}{rcl} \neg(C \sqcup D) & = & \neg C \sqcup \neg D \\ \neg(\exists R.C) & = & \forall R.\neg C \\ \mathcal{T} \models C \sqsubseteq D & iff. & C \sqcap \neg D \text{ is unsatisfiable wrt. the TBox } \mathcal{T} \end{array}$$

Answer:

There are two approaches to prove $C = D$ for two concepts C and D, one is by semantics, that is, we only need to prove that for any interpretation I, it must satisfy $C^I = D^I$. The other is by the tableau algorithm (try yourselves).

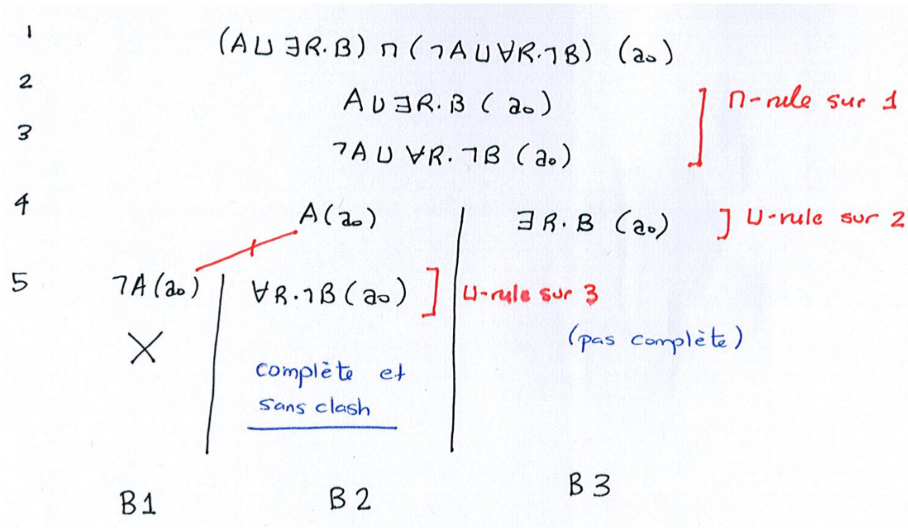


Figure 2: The second tableau construction

For the semantic approach, take $\neg(\exists R.C) = \forall R.\neg D$ for example. For any interpretation $I = (\Delta^I, \cdot^I)$, we need to prove $(\neg(\exists R.C))^I = (\forall R.\neg C)^I$. By the semantics of the logical connectors of $\mathcal{ALC}$, we have

$$
\begin{aligned}
(\neg(\exists R.C))^I &= \Delta^I \setminus (\exists R.C)^I \\
&= \Delta^I \setminus \{x \mid \text{ there is } y \in \Delta^I \text{ such that } (x, y) \in R^I\} \\
&= \{x \mid \text{ for all } y \in \Delta^I \text{ if } (x, y) \in R^I, y \notin C^I\} \\
&= \{x \mid \text{ for all } y \in \Delta^I \text{ if } (x, y) \in R^I, y \in (\neg C)^I\} \\
&= (\forall R.\neg C)^I
\end{aligned}
$$

Proof of $\mathcal{T} \models C \sqsubseteq D$ if and only if $C \sqcap \neg D$ is unsatisfiable wrt. the TBox $\mathcal{T}$:

$\mathcal{T} \models C \sqsubseteq D$ if and only if for any model I of $\mathcal{T}$, it satisfies $C \sqsubseteq D$, that is for any $x \in C^I$, we have $x \in D^I$, which means $x \notin (\neg D)^I$. That is, $x \notin (C \sqcap \neg D)^I$. So $C \sqcap \neg D$ is unsatisfiable wrt. $\mathcal{T}$. Otherwise, there must be at least one model I_0 and one individual $a_0 \in \Delta^{I_0}$ such that $a_0 \notin (C \sqcap \neg D)^{I_0}$, which leads to a contradiction.

Exercise n^o 8

Consider the following $\mathcal{ALC}$ ontology O:

$$
\begin{aligned}
\mathcal{T} &= \{A \sqsubseteq \forall R.B, B \sqsubseteq \neg F, E \sqsubseteq G, A \sqsubseteq D \sqcup \neg E, D \sqsubseteq \exists R.F, \exists R.\neg B \sqsubseteq G\} \\
\mathcal{A} &= \{A(a), R(a, b), F(b)\}
\end{aligned}
$$

Use the tableau algorithm to decide whether $\mathcal{T} \models A \sqsubseteq E$.

Answer. Try yourselves based on the third equivalence given in Exercise 7.

Exercice n^o 5

Read the W3C document about RDF (`http://www.w3.org/RDF`, `http://www.w3.org/TR/rdf-schema/`, and `http://www.w3.org/TR/rdf-schema/`) and answer the following questions:

1. Explain the following elements of the RDF vocabulary (presuming the usual namespace definitions).

$$\begin{array}{ll} (a) rdf : type & (b) rdf : about \\ (c) rdf : resource & (d) rdf : nil \\ (e) rdf : Property & (f) rdf : value \end{array}$$

2. Decide if the following propositions are true or false:
   (a) Blank nodes can stand for arbitrary resources.
   (b) URIs can stand for arbitrary resources.
   (c) Every blank node has an ID.
   (d) Two blank nodes with different IDs can stand for the same resource.
   (e) Two different URIs can stand for the same resource.
   (f) Blank nodes carrying the same ID that occur in several RDF documents must stand for the same resource.
   (g) URIs that occur in several RDF documents must stand for the same resource.
   (h) Two different Literals can never stand for the same value.
   (i) Two Literals with different datatype can never stand for the same value.
   (j) Blank nodes cannot occur in the predicate position of triples.

3. Consider the following RDF document and answer the following questions:
   (a) Describe in natural language the content of this document.
   (b) Draw the graph representation of the above document.
   (c) Translate the document into Turtle syntax. (Turtle specification: `http://www.w3.org/TR/turtle/`)

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:iswww="http://sw.edu/#"
>

<rdf:Description rdf:about="http://sw.edu/#germany">
  <rdf:type rdf:resource="http://sw.edu/#country" />
</rdf:Description>
<rdf:Description rdf:about="http://sw.edu/#capital_of">
  <rdf:type
   rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
  >

  <rdfs:domain rdf:resource="http://sw.edu/#city" />
  <rdfs:range rdf:resource="http://sw.edu/#country" />
</rdf:Description>
```

```
<rdf:Description rdf:about="http://sw.edu/#country">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  <rdfs:label xml:lang="de">Land</rdfs:label>
</rdf:Description>

<rdf:Description rdf:about="http://sw.edu/#berlin">
  <rdfs:label xml:lang="en">Berlin</rdfs:label>
  <rdf:type rdf:resource="http://sw.edu/#city" />
  <iswww:capital_of rdf:resource="http://sw.edu/#germany" />
</rdf:Description>

<rdf:Description rdf:about="http://sw.edu/#city">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class" />
  <rdfs:label xml:lang="de">Stadt</rdfs:label>
</rdf:Description>
</rdf:RDF>
```