14/06/2017 EX1

lunedì 4 gennaio 2021 09:32

Exercise 1 Given the following $\mathcal{ALC}$ TBox:

$$\begin{array}{rcl} A & \sqsubseteq & B \sqcup C \\ B & \sqsubseteq & \exists R.D \\ C & \sqsubseteq & \forall R.\neg D \\ A & \sqsubseteq & \forall R.E \\ D \sqcap E & \sqsubseteq & \neg B \end{array}$$

(a) tell whether the TBox $\mathcal{T}$ is satisfiable, and if so, show a model for $\mathcal{T}$;

(b) tell whether the concept A is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where the interpretation of A is non-empty;

(c) tell whether the concept $B \sqcap C$ is satisfiable with respect to $\mathcal{T}$, and if so, show a model for $\mathcal{T}$ where the interpretation of B is non-empty;

(d) given the ABox $\mathcal{A} = \{A(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A} \rangle$ entails the assertion $\exists R.E(a)$, explaining your answer.

a)

A Tbox is satisfiable if there exist a model for T. If exists an interpretation I that satisfies all axioms in the T.

Delta^I = {a}

A^I=B^I=C^I=D^I=E^I = r^I = empty set

A^I subseteq (B union C)^I ok
B^I subseteq (Exists r. D)^I ok
C^I subseteq (Forall r. not D)^I ok
A^I subseteq (Forall r. E)^I ok
(D and E)^I subseteq (not B)^I ok

I is a model for T

b)

A concept A is satisfiable w.r.t T if there exists an intepretation I in which this interpretation is a model of T and also the intepretation A^I is non empty.

If A is non empty B or C is non empy

If B is non empty means that Exists r. D should be non empty, so there will be al least one participation of an element of D in the role r.

Delta^I' = {a}

r^I' = {(a,a)}
B^I' = {}
C^I' = {a} (Forall r. not D)^I' = {a}
D^I' = {} (not D)^I' = {a}
E^I' = {a} (Forall r. E)^I = {a}

A^I'= {a}

The intepretation I' of A is non empty, so A is satisfiable w.r.t. T

c)

B is non empty if there exists at least one element of D that belong to r in the second position. C is non empty if there for participation of one element in the first position of the role r, the second element should belong to not D.
There is a contradiction because we cannot have the same element that belong to C and to B, because if we cannot have the same element in not D and D. For example in D^I= {a}
The interpretation of (not D)^I = {b}. r^I ={(a,a)} (exists r. D) = {a} instead (forall r. not D)^I = {b}

d)

Not ((Exists r. E)(a)) -> NNF : (Forall r. not E)(a)

A_0 = {A(a), (Forall r. not E)(a))

C_GCI = (not A union B union C) and (not B union exists r. D) and (not C union forall r. not D) and (not A union forall r. E) and (not (D and E) union not B)

NNF: not (D and E) -> not D union not E

C_GCI = (not A union B union C) and (not B union exists r. D) and (not C union forall r. not D) and (not A union forall r. E) and (not D union not E union not B)

(C_GCI-rule) A_1 = A_0 union {((not A union B union C) and (not B union exists r. D) and (not C union forall r. not D) and (not A union forall r. E) and (not D union not E union not B))(a)}
(and- rule) A_2 = A_1 union {(not A union B union C)(a),(not B union exists r. D)(a),(not C union forall r. not D)(a), (not A union forall r. E)(a),(not D union not E union not B)(a)}

(or-rule) A_3= A_2 union {not A(a)} - CLASH
A_4= A_2 union {B(a)}
A_5= A_2 union {C(a)}
(or-rule) A_6=A_4 union {not B} - CLASH
A_7 = A_4 union {exists r. D}
(Exist-rule) A_8 = A_7 union {D(x), r(a,x)}
(C_GCI-rule) A_9 = A_8 union {((not A union B union C) and (not B union exists r. D) and (not C union forall r. not D) and (not A union forall r. E) and (not D union not E union not B))(x)}
(and- rule) A_10 = A_9 union {(not A union B union C)(x),(not B union exists r. D)(x),(not C union forall r. not D)(x), (not A union forall r. E)(x),(not D union not E union not B)(x)}
(Forall- rule) A_11 = A_10 union {not E(x)}
(or-rule) A_12 = A_11 union {not A(a)} - CLASH
A_13 = A_11 union {(forall r. E)(a)}
(forall-rule) A_14 = A_13 union{E(x)} - CLASH
(or-rule) A_15 = A_5 union {not C(a)} - CLASH
A_16 = A_5 union {(forall r. not D)(a)}
(or-rule) A_17 = A_16 union {not B(a)} ---> open and complete
A_18 = A_16 union {(exist r. D)(a)}

Tableau algorithm return True, and this means that (Exist r. E)(a) is false, it is not entailed by KB