18/2/2019 EX1

mercoledì 6 gennaio 2021 12:41

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

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

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

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

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

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

a)

Delta^I = {a}
A^I=F^I = B^I =H^I=C^I=D^I=E^I =G^I = r^I= empty set

I is a model for T, T is satisfiable

b)
C is satisfible, if I is a model for T and C^I is non empty

C^I is non empty if D^I or E^I non empty

Delta^I' = {a}
D^I' = F^I= A^I' = H^I = empty set
C^I' = {a}
E^I' = {a}
G^I' = {a}
r^I' = {(a,a)}

(Exists r. G)^I' = {a}

I' is a model for T and C^I' is non empty, C is satisfiable

c)

If D is non empty means that F should be non empty and Forall r. A is non empty.
A is non empty if not F is non empty.
Then If B is non empty, also B and F shoud non empty, and this means that Exists r. H should be non empty but H is non empty if F is non empty
So at en we need to have Forall r. not F and Exists r. F that should be non empty. They are non empty but the problem is that the intersection between them should be always empty because we cannot have an element of Forall r.not that it is equal to element of Exists r. F because in F and not F there opposite element.

For example we have
Delta^I = {a,b}
F^I = {a}
(Not F)^I = {b}
r^I = {(a,b), (b,a)}

(Exists r. F) ^I = {b}
(Forall r. not F)^I = {a}

((Exists r. F) and (Forall r. not F))^I = empty set

d)

A0= {B(a), C(a), not E(a)}

C_GCI= (not A or not F) and (not (B and F) or Exists r. H) and (not C or D or E) and (not D or (F and Forall r. A) and (not E or Exists r. G) and (not H or F)

NNF: not(B and F) -> not B or not F
C_GCI= (not A or not F) and (not B or not F or Exists r. H) and (not C or D or E) and (not D or (F and Forall r. A) and (not E or Exists r. G) and (not H or F)

(C_GCI -rule) A1 = A0 union {((not A or not F) and (not B or not F or Exists r. H) and (not C or D or E) and (not D or (F and Forall r. A) and (not E or Exists r. G) and (not H or F)) (a)}
(and-rule) A2 = A1 union { (not A or not F)(a), (not B or not F or Exists r. H)(a), (not C or D or E)(a), (not D or (F and Forall r. A)(a), (not E or Exists r. G)(a), (not H or F)(a)}
(or-rule) A3 = A2 union {not B(a)} -CLASH
A4 = A2 union {not F(a)}
A5 = A2 union {(Exists r. H)(a)}
(or-rule) A6 = A4 union {(F and Forall r. A)(a)}
A7 = A4 union {not D(a)}
(and-rule) A8 = A6 union {F(a), (Forall r. A)(a)} - CLASH
(or-rule) A9 = A7 union {not C(a)} - CLASH
A10 = A7 union {D(a)} - CLASH
A11 = A8 union {E(a)} - CLASH
(or-rule) A12 = A5 union {not C(a)} - CLASH
A13 = A5 union {E(a)} - CLASH
A14 = A5 union {D(a)}
(or-rule) A15 = A14 union {not D(a)} - CLASH
A16 = A14 union {(F and Forall r. A)(a)}
(and-rule) A17 = A16 union {F(a), (Forall r. A)(a)}
(or-rule) A18 = A17 union {not F(a)} - CLASH

A19 = A17 union {not A(a)}
(Exists-rule) A20 = A19 union{r(a,x), H(x)}
(C_GCI -rule) A21 = A20 union {((not A or not F) and (not B or not F or Exists r. H) and (not C or D or E) and (not D or (F and Forall r. A) and (not E or Exists r. G) and (not H or F)) (x)}
(and-rule) A22 = A21 union { (not A or not F)(x), (not B or not F or Exists r. H)(x), (not C or D or E)(x), (not D or (F and Forall r. A)(x), (not E or Exists r. G)(x), (not H or F)(x)}
(or-rule) A23 = A22 union {not H(x)} - CLASH
A24 = A22 union {F(x)}
(or-rule) A25 = A24 union {not F(x)} - CLASH
A26 = A24 union {not A(x)}
(Forall-rule) A27 = A26 union {A(x)} - CLASH

All Aboxes are closed, tableau return false, the instance checking problem in true