# 13/6/2018 EX1

venerdì 1 gennaio 2021 16:42

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

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

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

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

(c) given the ABox $\mathcal{A}' = \{D \sqcap F(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A}' \rangle$ is satisfiable (consistent), and if so, show a model for $\langle \mathcal{T}, \mathcal{A}' \rangle$;

(d) given the ABox $\mathcal{A}'' = \{D(a)\}$, tell whether the knowledge base $\langle \mathcal{T}, \mathcal{A}'' \rangle$ entails the assertion $A(a)$, explaining your answer.

a)

A Tbox is satisfiable if there exists a model for T, so if there exists an interpretation that satisfies all axioms.

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

I is a model for T

b)

B is satisfiable if there is an interpretation I in which B is non empty and I is a model for T

B should be non empty, so not F, C and Forall r. F should be non empty.
C is non empty if E is non empty and E is non empty if there is at least one element of B that belong to the role r. But B we say that B is non empty if we have that not F is non empty.
So both Forall r. F and Exist r not F should not contain the same element

.
B is non satisfiable

c)
A' = {(D and F)(a)}

(and-rule) A0 = A' union {D(a),F(a)}

C_GCI = (not A or not F) and (not B or not F) and (not B or C) and (not B or forall r. F) and (not C or E) and (not D or A or B) and (not D or Exists r. A) and (not E or Exists r. B)
(C_GCI-rule) A1 = A0 union {((not A or not F) and (not B or not F) and (not B or C) and (not B or forall r. F) and (not C or E) and (not D or A or B) and (not D or Exists r. A) and (not E or Exists r. B))(a)}
(and-rule) A2 = A1 union {(not A or not F)(a), (not B or not F)(a), (not B or C)(a), (not B or forall r. F)(a), (not C or E)(a), (not D or A or B)(a), (not D or Exists r. A)(a), (not E or Exists r. B)(a)}
(or-rule) A3 = A2 union {not D(a)} - CLASH
A4 = A2 union {A(a)}
A5 = A2 union {B(a)}
(or-rule) A6 = A4 union {not A(a)} - CLASH
A7 = A4 union {not F(a)} - CLASH

(or-rule) A8 = A5 union {not B(a)} - CLASH
A9 = A5 union {not F(a)} - CLASH

KB is unsatisfiable because tableau return false

d)

This is a problem of instance checking in which we want to verifies if that concept assertion is entailed by KB. We want to apply the tableau rule, so we need to reduct our instance checking problem of a satisfiability problem by do the negation of this concept assertion
The we add to our Abox and apply the tableau rules. If there is at least one Abox produced by tableau that is open and complete, the tableau return true and so the instance checking problem is false. Otherwise, if all Aboxes are closed, so there is an atomic contraddiction, the tableau algorithm return false and the instance checking problem is true.

A0 = {D(a),not A(a)}
C_GCI = (not A or not F) and (not B or not F) and (not B or C) and (not B or forall r. F) and (not C or E) and (not D or A or B) and (not D or Exists r. A) and (not E or Exists r. B)
(C_GCI-rule) A1 = A0 union {((not A or not F) and (not B or not F) and (not B or C) and (not B or forall r. F) and (not C or E) and (not D or A or B) and (not D or Exists r. A) and (not E or Exists r. B))(a)}
(and-rule) A2 = A1 union {(not A or not F)(a), (not B or not F)(a), (not B or C)(a), (not B or forall r. F)(a), (not C or E)(a), (not D or A or B)(a), (not D or Exists r. A)(a), (not E or Exists r. B)(a)}
(or-rule) A3 = A2 union {not D(a)} - CLASH
A4 = A2 union {A(a)} - CLASH
A5 = A2 union {B(a)}
(or-rule) A6 = A5 union {not B(a)} - CLASH
A7 = A5 union {not F(a)}
(or-rule) A8 = A7 union {not B(a)}- CLASH
A9 = A7 union {C(a)}
(or-rule) A10 = A9 union {not C(a)} - CLASH
A11 = A9 union {E(a)}
(or-rule) A12 = A11 union {not E(a)} - CLASH
A13 = A11 union {(Exists r. B)(a)}
(or-rule) A14 = A13 union {not B(a)} - CLASH
A15 = A13 union {(Forall r. F)(a)}
(or-rule) A16 = A15 union {not D(a)} - CLASH
A17 = A15 union {(Exists r. A)(a)}
(Exists -rule) A18 = A17 union {B(x),r(a,x)}
(C_GCI-rule) A19 = A18 union {((not A or not F) and (not B or not F) and (not B or C) and (not B or forall r. F) and (not C or E) and (not D or A or B) and (not D or Exists r. A) and (not E or Exists r. B))(x)}
(and-rule) A20 = A19 union {(not A or not F)(x), (not B or not F)(x), (not B or C)(x), (not B or forall r. F)(x), (not C or E)(x), (not D or A or B)(x), (not D or Exists r. A)(x), (not E or Exists r. B)(x)}
(Forall-rule) A21 = A20 union {F(x)}
(or-rule) A22 = A21 union {not F(x)} - CLASH
A23 = A21 union {not B(x)} - CLASH

All Aboxes are close, tableau return false, so the instance checking problem is true