1. **Check whether the above instantiation, once completed, is correct, and explain why it is or it is not. 2. Express in FOL the following queries and evaluate them over the completed instantiation:**

(a) Return the sailors that have been on board of a boat which has been in a harbor where a tag boat works in.

Sailor(x) and Exists y. onboard(x, y) and Exists z. been(y, z) and Exist w. worksin(w, z)

(b) Check whether there exists a harbor in which there have been at least two tag boats.

Exists x. Harbor(x) and Exists y, y'. Tugboat(y) and TugBoat(y') and been(y, x) and been(y',x) and y!=y'

(c) Return the sailors that have been in all harbors.

Sailor(x) and Forall y. Harbor(y) implies Exists z. onboard(x, z) and been(z, y)

Exercise 3. Model check the Mu-Calculus formula νX.μY.((a ∧ [next]X) ∨ ([next]Y)) and the CTL formula EF(AG(a ⊃ EXEX¬a)) (showing its translation in Mu-Calculus) against the following transition system:

Φ = νX.μY.((a ∧ [next]X) ∨ ([next]Y))

[| X0 |] = {0, 1, 2, 3, 4}
[| X1 |] = [| μY.((a ∧ [next]X) ∨ ([next]Y)) |]

[| Y00 |] = { }
[| Y01 |] = [| (a ∧ [next]X0) ∨ ([next]Y00) |] =
= [| a |] inter PreA(next, X0) U PreA(next, [| Y00 |]) =
= {0, 1, 4} inter {0, 1, 2, 3, 4} U {} = {0, 1, 4}

[| Y02 |] = [| (a ∧ [next]X0) ∨ ([next]Y01) |] =
= [| a |] inter PreA(next, X0) U PreA(next, [| Y01 |]) =
= {0, 1, 4} inter {0, 1, 2, 3, 4} U {0, 3, 4} = {0, 1, 3, 4}
[| Y03 |] = [| (a ∧ [next]X0) ∨ ([next]Y02) |] =
= [| a |] inter PreA(next, X0) U PreA(next, [| Y02 |]) =
= {0, 1, 4} inter {0, 1, 2, 3, 4} U {0, 3, 4} = {0, 1, 3, 4}

[| Y03 |] = [| Y02 |] -> LFP = {0, 1, 3, 4}

[| X2 |] = [| μY.((a ∧ [next]X) ∨ ([next]Y)) |]

[| Y10 |] = { }

[| Y11 |] = [| (a ∧ [next]X1) ∨ ([next]Y10) |] =

= [| a |] inter PreA(next, X1) U PreA(next, [| Y10 |]) =

= {0, 1, 4} inter {0, 3, 4} U {} = {0, 4}

[| Y12 |] = [| (a ∧ [next]X1) ∨ ([next]Y11) |] =

= [| a |] inter PreA(next, X1) U PreA(next, [| Y11 |]) =

= {0, 1, 4} inter {0, 3, 4} U {3, 4} = {0, 3, 4}

[| Y13 |] = [| (a ∧ [next]X1) ∨ ([next]Y12) |] =

= [| a |] inter PreA(next, X1) U PreA(next, [| Y12 |]) =

= {0, 1, 4} inter {0, 3, 4} U {3, 4} = {0, 3, 4}

[| Y13 |] = [| Y12 |] = Found LFP = {0, 3, 4}

[| X3 |] = [| μY.((a ∧ [next]X) ∨ ([next]Y)) |]

[| Y20 |] = { }

[| Y21 |] = [| (a ∧ [next]X2) ∨ ([next]Y20) |] =

= [| a |] inter PreA(next, X2) U PreA(next, [| Y20 |]) =

= {0, 1, 4} inter {3, 4} U {} = {4}

[| Y22 |] = [| (a ∧ [next]X2) ∨ ([next]Y21) |] =

= [| a |] inter PreA(next, X2) U PreA(next, [| Y21 |]) =

= {0, 1, 4} inter {3, 4} U {3} = {3, 4}

[| Y23 |] = [| (a ∧ [next]X2) ∨ ([next]Y22) |] =

= [| a |] inter PreA(next, X2) U PreA(next, [| Y22 |]) =

= {0, 1, 4} inter {3, 4} U {3} = {3, 4}

[| Y23 |] = [| Y22 |] = LFP = {3, 4}

[| X4 |] = [| μY.((a ∧ [next]X) ∨ ([next]Y)) |]

[| Y30 |] = { }

[| Y31 |] = [| (a ∧ [next]X3) ∨ ([next]Y30) |] =

= [| a |] inter PreA(next, X3) U PreA(next, [| Y30 |]) =

= {0, 1, 4} inter {3} U {} = { }

[| Y32 |] = [| (a ∧ [next]X3) ∨ ([next]Y31) |] =
= [| a |] inter PreA(next, X3) U PreA(next, [| Y31 |]) =
= {0, 1, 4} inter {3} U {3} = { }

[| Y32 |] = [| Y31 |] = { }

[| X5 |] = [| X4 |] = GFP { }

1 in PHI? NO.

Decomposing CTL formula EF(AG(a ⊃ EX EX¬a))

Alpha = EX¬a
Beta = EX alpha
Gamma = a ⊃ Beta
Delta = AG(Gamma)
Epsilon = EF(Delta)

T(Alpha) = <Next> ¬a
T(Beta) = <Next> Alpha
T(Gamma) = ¬a OR Beta
T(Delta) = νX. Gamma AND [Next] X
T(Epsilon) = μX. Delta OR <Next> X

[|Alpha|] = [|<Next> ¬a |] = [| PreE(next, [|not a|])|] = {0, 1, 2}
[|Beta|] = [|<Next> Alpha|] = PreE(next, [|Alpha|]) = {0, 1, 2, 4}
[|Gamma|] = [|¬a OR Beta |] = [|¬a |] U [|Beta|] = {2, 3} U {0, 1, 2, 4} =
= {0, 1, 2, 3, 4}

[|Delta|] = [|νX. Gamma AND [Next] X|]
[|X0|] = {0, 1, 2, 3, 4}
[|X1|] = [|Gamma |] inter PreA(next, [|X0|]) = {0, 1, 2, 3, 4} inter {0, 1, 2, 3, 4} = {0, 1, 2, 3, 4}

[|Epsilon|] = [|μX. Delta OR <Next> X|]
[|X0|] = {}
[|X1|] = [|Delta |] U preE(next, X0) = {0, 1, 2, 3, 4} U {} = {0, 1, 2, 3, 4}

1 in [|Epsilon|]? YES.

Exercise 5. Given the following boolean conjunctive queries (with *a* constant):

q1() :- e(a,y),e(y,y),e(y,a)
q2() :- e(a,y),e(y,z),e(z,w),e(w,w),e(w,z),e(z,y),e(y,a)

check whether q1 is contained into q2, explaining the technique used and, in case of containment, showing the homomorphism between the canonical databases.

Check if q1() subseteq q2(). How?

Recognition problem:

Forall I, alpha models Forall x. Q1(x) implies q2(x) is VALID

Building Iq1:

Iq1= {Delta, E, C}
Delta={a, y}
E= {e(a,y),e(y,y),e(y,a) }
C={a}

Tabula form Iq1:

{(a,y)
(y,y)
(y,a)}

Assingment function alpha for non distinguished variables of q2:

alpha(y) = y
alpha(z) = a
alpha(w) = y

This is a satisfying assignment.

From CM Theorem:

Iq1 models q2 iff Exists h. Iq2 ->h Iq1

To check homomorphism, build the canonical Interpretation of Iq2:
Iq2={Delta, e, c}
Delta={ a, y, z, w}
E={ e(a,y),e(y,z),e(z,w),e(w,w),e(w,z),e(z,y),e(y,a)}
C={a}

Tabula form: { (a,y)
(y,z)
(z,w)
(w,w)
(w,z)
(z,y)
(y,a) }

H(a) = h(cIq2) = a
H(y) = alpha(y) = y
H(z) = alpha(z) = a
H(w) = alpha(w) = y

CM theorem states that this is a homomorphism, we can prove that it is true, if the relation is maintained by the mapping of Iq2 -> Iq1:

(a, y) in Iq2 -> (h(a), h(y)) = (a, y) in Iq1
(y, z) in Iq2 -> (h(y), h(z)) = (y, a) in Iq1
(z, w) in Iq2 -> (h(z), h(w)) = (a, y) in Iq1
(w, w) in Iq2 -> (h(w), h(w)) = (y, y) in Iq1
(w, z) in Iq2 -> (h(w), h(z)) = (y, a) in Iq1
(z, y) in Iq2 -> (h(z), h(y)) = (a, y) in Iq1
(y, a) in Iq2 -> (h(y), h(a)) = (y, a) in Iq1