Concordia University Department of Computer Science and Software Engineering

SOEN 331 Section S: Formal Methods for Software Engineering

Assignment 1

Mohammad Ali Zahir - 40077619

Marwa Khalid - 40155098

October 1, 2022

Date of Submission: October 10, 2022

Contents

1 Problem 1: Predicate Logic 1 (10 pts)

1.1 Description:

In the domain of all people in the room, consider the predicate *received_request(a,b)* that is intepreted as

"[person] a has received a request from [person] b to connect on some social platform"

1. How are the following two expressions translated into plain English? Are the two expressions logically equivalent?

   - $\forall a\ \exists b$ *received_request(a, b).*
   - $\exists b\ \forall a$ *received_request(a, b).*

   Solution:

   The first statement $\forall\ a\ \exists\ b$ *received_request(a, b).* reads that every person a has received a request from one person b.

   The second statement $\exists\ b\ \forall\ a$ *received_request(a, b).* reads that there exists a person b which has a received a request from all people a.

   In terms of logical equvialency, we would need to determine if the truth values for both of these statements are the same. For the first statement,

   while it is possible that every person a has received a request from one person b, it is highly unlikely that in the second statement, that one person b has received

   a request from every person from a. **Hence, both of these statement are NOT logically equivalent**.

2. Discuss in detail whether we can we claim the following:

   $$\forall a\ \exists b\ \textit{received\_request(a, b)} \rightarrow \exists b\ \forall a\ \textit{received\_request(a, b).}$$

   Solution:

   The statement $\forall a\ \exists b$ *received_request(a, b)* $\rightarrow \exists b\ \forall a$ *received_request(a, b).* is **FALSE**.

   The first predicate states that all the people from a have received a request from each person b. While this does mean that everyone person a received a request from a person b, it does not neccessarily mean that all the a people asked the same b people. Every a person could have received a request from a different b person. Predicate 2, however, states that there exists a person b who has received a request from every person a, which we have proved from the above statement is **FALSE**.

3. Discuss in detail whether we can we claim the following:

   $$\exists b\ \forall a\ \textit{received\_request(a, b)} \rightarrow \forall a\ \exists b\ \textit{received\_request(a, b)}.$$

   Solution:

   The statement $\exists b\ \forall a$ *received_request(a, b)* $\rightarrow \forall a\ \exists b$ *received_request(a, b).* is **TRUE**. The first predicates states that there is exists one person b who has a received a request from every person a. The second predicate states that every person a has received a request from atleast one person b which we see from the first predicate to be true, hence we can claim this statement to be **TRUE**.

4. How are the following two expressions translated into plain English? Are the two expressions logically equivalent?

   - $\forall b\ \exists a$ *received_request(a, b).*
   - $\exists a\ \forall b$ *received_request(a, b).*

   Solution:

   The first statement $\forall b\ \exists a$ *received_request(a, b).* reads that every person b has received a request from one such person a. The second statement $\exists a\ \forall b$ *received_request(a, b).* states that there exists a person a which has a received from every person b. To make sure that these statements are logically equivalent, we would need to determine if the truth values for these statements are the same. While in the first statement, every person b does receive a request from one person a, we have no way to make sure that every person b get the request from the same person a. If that statement would hold, then we could say that there would exist a person a who receives a request from every person b, but since this is not the case **Both of these statements are NOT logically equivalent**.

2 Problem 2: Predicate Logic 2 (10 pts)

2.1 Description:

Given the subject "being a person" and the predicate "being bad", consider the list of propositions below:

1. "There are some nice people"
2. "There are no nice people"
3. "Everybody is bad"
4. "Some people are bad"
5. "Everybody is nice"
6. "Some people are not nice"

Associate each of the propositions below to one of the standard forms of categorical propositions.

Solution:

We can assume in this case that being a person (or people) would be the subject S, denoted by P(x) and the being bad would be the predicate P, denoted by Q(x). Let:

1. "There are some nice people" = This is of the form some S are not P. Translated to propositon form, this would give us $\exists\, x, (P(x) \wedge \neg Q(x))$, **which is O form**.
2. "There are no nice people" = This is of the form of all S are P. Translated to propositon form, this would give us $\forall\, x, (P(x) \rightarrow Q(x))$, **which is A form**.
3. "Everybody is bad" = This is of the form of no S are P. Translated to proposition form, this would give us $\forall\, x, (P(x) \rightarrow \neg Q(x))$, **which is E form**.
4. "Some people are bad" = This is of the form of some S are P. Translated to proposition form, this would give us $\exists\, x, (P(w) \wedge Q(x))$, **which is I form**.

5. "Everybody is nice" = This is of the form of no S are P. this would give us $\forall x, (P(x) \to \neg Q(x))$, **which is E form**.

6. "Some people are not nice" = This is of the form of some S are P. Translated to proposition form, this would give us $\exists x, (P(w) \wedge Q(x))$, **which is I form**.

3 Problem 3: Unordered and ordered structures (15 pts)

3.1 Description:

Consider the following two sets:

- OS = {*MacOS, Linux, BSD, Windows, Unix*}, and
- *My_OS* = {*BSD, Unix*}.

Answer the following questions:

1. Is the following declaration acceptable: $My\_OS \, \mathbb{P}OS$? Explain.

   Solution:

2. Is $\mathbb{P}OS$ a legitimate type? Explain.

   Solution:

3. What does the following statement signify? My OS : OS. Is the statement acceptable? Explain.

   Solution:

4. Is $MacOS\mathbb{P}OS$? Explain.

   Solution:

5. Is OS a legitimate type?

   Solution:

6. Is $\{\} \in \mathbb{P}OS$? Explain.

   Solution:

7. Is $\{Linux,\ BSD\} \in \mathbb{P}OS$? Explain.

   Solution:

8. Is $\{\{\}\} \in \mathbb{P}OS$?

   Solution:

9. Is $\{\} \in OS$? Explain.

   Solution:

10. If we define variable $My\_Computer : \mathbb{P}OS$, is $\{\}$ a legitimate value for variable $My\_Computer$? Explain.

    Solution:

11. If we stated that $My\_Computer = \{Windows\}$, would the statement $My\_Computer$ make an atomic variable?

    Solution:

12. Is $\{\{BSD, MacOS\}\} \subset \mathbb{P}OS$? Explain.

    Solution:

13. Is $My\_OS \subset \mathbb{P}OS$? Explain.

    <u>Solution:</u>

14. Is $\{\{BSD, MacOS\}\} \in \mathbb{P}OS$?

    <u>Solution:</u>

4 Problem 4: Relational calculus 1 (15 pts)

4.1 Description:

Consider a system that associates active flights to airlines. The requirements of the system are as follows:

- Flights are unique.
- Each flight is associated to a single airline, e.g. AA333 is an American Airlines flight.
- The system can support new flights to be associated to an existing airline, or existing flights to be deleted.
- Airlines can have several active flights at any point in time.

We introduce types *Flight* and *Airline*. The model of the system is captured by variable map, as shown below:

$$
\begin{array}{l}
maps = \\
\quad \{ \\
\quad\quad AAA333 \mapsto American\ Airlines, \\
\quad\quad AY29 \mapsto Finnair, \\
\quad\quad TS261 \mapsto Air\ Transat, \\
\quad\quad TS765 \mapsto Air\ Transat \\
\quad \}
\end{array}
$$

1. Is *map* a binary relation? Explain.

   Solution:

2. Is *map* a function? Explain and if Yes, determine the type of the function.

   Solution:

3. Define the precondition for operation *add*, that adds a new flight-airline pair.

   Solution:

   For Questions 4 and 5 assume the presence of the above precondition:

4. Provide two alternative definitions for the core functionality of operation *add*.

   Solution:

   For Questions 6 and 7 assume that the above precondition is removed:

5. What would be the result of calling operation *add* with $flight? = TS765$, and $airline? = American\ Airlines$?

   Solution:

6. What would be the result of calling *add* with

$$flight? = AA333,$$
$$airline? = AirCanada,$$

   Solution:

7. Under what conditions, if any, can *set union* serve as a mechanism to successfully add a new record into the database table? What error could possibly occur?

Solution:

8. Provide a definition for the core functionality of operation **delete** that erases a flight from *map*, given the flight number.

   Solution:

5 Problem 5: Relational calculus 2 (25 pts)

5.1 Description:

Consider the following binary relation:

$$airplanes : Model \leftrightarrow Manufacturer$$

where

$$
\begin{array}{l}
airplanes = \\
\quad \{ \\
\qquad A320 \mapsto Airbus, \\
\qquad A330 \mapsto Airbus, \\
\qquad A350 \mapsto Airbus, \\
\qquad A380 \mapsto Airbus, \\
\qquad 737 \mapsto Boeing, \\
\qquad 747 \mapsto Boeing, \\
\qquad Superjet100 \mapsto Sukhoi, \\
\qquad C919 \mapsto Comac, \\
\qquad Global7500 \mapsto Bombardier, \\
\qquad Global8000 \mapsto Bombardier, \\
\qquad E170 \mapsto Embraer, \\
\qquad E175 \mapsto Embraer \\
\quad \}
\end{array}
$$

1. What is the value of the following expression:

$$\{A330, 747\} \lhd airplanes$$

<u>Solution:</u>

2. What is the value of the following expression:

$$airplanes \rhd \{Comac, Embraer\}$$

<u>Solution:</u>

3. What is the value of the following expression:

$$\{A320, A330, A350, E170\} \lhd airplanes$$

<u>Solution:</u>

4. What is the value of the following expression:

$$airplanes \rhd \{Airbus, Boeing\}$$

<u>Solution:</u>

5. What is the value of the following expression:

$$airplanes \oplus \{Su\_80 \mapsto Sukhoi\}$$

<u>Solution:</u>