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:

   $\forall a\ \exists b$ *received_request(a, b)* $\rightarrow \exists b\ \forall a$ *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:

   $\exists b\ \forall a$ *received_request(a, b)* $\rightarrow \forall a\ \exists b$ *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 $\forall\, x, (P(x) \rightarrow \neg Q(x))$, **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:

   Since the set *My_OS* contains all the values in *OS* set, we can say that this is an acceptable declaration.

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

   Solution:

   This is a legitimate type, since *OS* can assume any variable from the set *My_OS*.

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

   Solution:

   We can interpret this is as *'The variable My_OS can assume any value which is contained in the set OS'*

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

   Solution:

   Since POS is a powerset, and a powerset can only contain a set as it's element, **this statement is FALSE, since *MacOS* is an atomic element and not a set.**

5. Is OS a legitimate type?

   Solution:

   Since we do not have a restriction on what the type can be, **we can say yes that this is a legitimate type**.

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

   Solution:

   Since this is considered as a empty set, and the powerset all the subsets including the empty one, **this statement is TRUE**.

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

   Solution:

   Since the subset of the set $\{Linux,\ BSD\}$ is contained in the powerset $\mathbb{P} OS$, **we can say that this statement is TRUE**.

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

   Solution:

   As this is translated as the set of the empty set, which is basically the set of a set. This definition is not contained in a powerset, **this statement is FALSE**.

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

   Solution:

   Since we are talking about an empty set being part of a non-powerset, the OS set does not need to contain all the subsets like the powerset, **this statement is FALSE**.

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

    Solution:

With that statement, we are saying that $My\_Computer$ is a variable for the type $\mathbb{P}OS$, and that powerset contains all the subsets of the variable which also includes which also includes the empty set like in this case, **this value is legitimate**.

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

    Solution:

    No it would not since $\{Windows\}$ is a set.

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

    Solution:

    Since this set is an element of the set, **hence this statement is TRUE**.

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

    Solution:

    The set My_OS does not contain any sets as it's elements, **hence this statement is FALSE**.

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

    Solution:

    Since this set is an element of the set My_OS, **this statement is TRUE**.

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:

$map =$

$\{$

$AAA333 \mapsto American\ Airlines,$

$AY29 \mapsto Finnair,$

$TS261 \mapsto Air\ Transat,$

$TS765 \mapsto Air\ Transat$

$\}$

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

   Solution:

   Since this relation relates the the elemments of one set (flight), which is the domain, with the elements of another set airplane which is the codomain. **The relation map is hence considered a binary relation**.

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

   Solution:

   We could say that this is a function since all elements of the domain are unique and are linked to exactly one member of the codomain.

   In terms of what kind of function this is, we cannot say this is one-to-one function since both TS261 and TS765 are both linked to the Air Transat airline. Since this confirms that the fucntion is not one-to-one, we can also confirm that this is not a bijective function since for a function to be bijective, it has to be onto and one-to-one. **Hence, we can say that this function is onto, since all elements of the codomain (airline), is mapped by atleast one member of the domain (flight)**.

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

Solution:

Intially we would need to check if the ordered pair exists in maps. To check this, we would need to see that the flight is not contained in the database.

If that doesn't exist, then we can add the new ordered pair. The translated statement for the add would be:

flight* $\notin$ *dom map

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:

   To add the operation, we can either do a union or we can do the insertion operator. The statements needed for those would look like the following:

   - *map' = map=* $\cup$ $\{$*flight* $\mapsto$ *airline*$\}$
   - map' = map= $\oplus$ $\{$*flight* $\mapsto$ *airline*$\}$

   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:

   Since the flight TS765 already exists in the domain of maps, **this operation will give us a failure**.

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

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

   Solution:

Since this flight does not exist in the domain of maps, we would be able to add this ordered pair.

We would the use the insertion operator, which would look like:

$$\begin{aligned}
&\textit{map'} = \textit{map} \oplus \{AA333 \mapsto \textit{AirCanada}\} = \quad \{ \\
&\qquad AA333 \mapsto \textit{Air Canada}, \\
&\qquad AAA333 \mapsto \textit{American Airlines}, \\
&\qquad AY29 \mapsto \textit{Finnair}, \\
&\qquad TS261 \mapsto \textit{Air Transat}, \\
&\qquad TS765 \mapsto \textit{Air Transat} \\
&\quad \}
\end{aligned}$$

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:

   Set Union is used to add a new record. It would first check if the record exists by checking if the domain exists in the database already.

   If such an element does not exist in the database, we add the new ordered pair. However, since this is a union operation, it will still add the ordered pair even if the element does exists, which would mean we would have duplicates in our databse.

   This will make it that we will have two elements in the same domain (which is flight), which violates the condition that all flights are unique.

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

   Solution:

   Intially we would need to check if the ordered pair exists in maps. We would need to see that the flight is contained in the databse.

   If that does exist, then we can delete the existing ordered pair. The translated statement for the delete would be:

   flight $\in$ ***dom map***

5 Problem 5: Relational calculus 2 (25 pts)

5.1 Description:

Consider the following binary relation:

$$airplanes : Model \leftrightarrow Manufacturer$$

where

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

1. What is the value of the following expression:

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

Solution:

This is translated to *" select all the pairs of model whose domain is A330 and 747"*. The result that we get from this is

$$\begin{aligned} \{A330, 747\} \lhd \mathit{airplanes} = \quad \{ & \\ & A330 \mapsto \mathit{Airbus}, \\ & 747 \mapsto \mathit{Boeing} \\ \} & \end{aligned}$$

2. What is the value of the following expression:

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

Solution:

This is translated to *" select all the pairs of manufacturers whose range is Comac and Embraer"*. The result that we get from this is

$$\begin{aligned} \mathit{airplanes} \rhd \{\mathit{Comac}, \mathit{Embraer}\} = \quad \{ & \\ & C919 \mapsto \mathit{Comac}, \\ & E170 \mapsto \mathit{Embraer}, \\ & E175 \mapsto \mathit{Embraer} \\ \} & \end{aligned}$$

3. What is the value of the following expression:

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

Solution:

This is translated to *" remove all the pairs of models whose domain is A320, A330, A350 and E170"*. The result that we get from this is

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

This is called domain substraction. While this demonstrated what the effect will be, for the effect to properly take place, we must do *airplanes'* = $\{$*A320,A330,A350, E170* $\}$ $\lhd$*airplanes*

4. What is the value of the following expression:

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

Solution:

This is translated to *" remove all the pairs of models whose range is Airbus and Boeing"*. The result that we get from this is

$$
\begin{aligned}
&\textit{airplanes} \rhd \{\textit{Airbus}, \textit{Boeing}\} = \quad \{\\
&\quad \textit{Superjet}100 \mapsto \textit{Sukhoi},\\
&\quad C919 \mapsto \textit{Comac},\\
&\quad \textit{Global}7500 \mapsto \textit{Bombardier},\\
&\quad \textit{Global}8000 \mapsto \textit{Bombardier},\\
&\quad E170 \mapsto \textit{Embraer},\\
&\quad E175 \mapsto \textit{Embraer}\\
&\}
\end{aligned}
$$

This is called range substraction. While this demonstrated what the effect will be, for the effect to properly take place, we must do *airplanes'* = airplanes $\rhd\{\textit{Airbus}, \textit{Boeing}\}$

5. What is the value of the following expression:

$$\textit{airplanes} \oplus \{\textit{Su\_}80 \mapsto \textit{Sukhoi}\}$$

Solution:

When the symbol $\oplus$ is involved, we can either assume we are going to insert or modify in the database

In this case, however, we do not have *Su_80* in the database already, hence this has to be an insertion. For an insertion to properly happen in the DB,

We would need to *airplane'* = airplanes $\oplus\{\textit{Su\_}80 \mapsto \textit{Sukhoi}\}$, but for the sake of simplicity for this problem, We would have

$$
\begin{aligned}
&\mathit{airplanes} \oplus \{\mathit{Su\_80} \mapsto \mathit{Sukhoi}\} = \quad \{\\
&\quad \mathit{A320} \mapsto \mathit{Airbus},\\
&\quad \mathit{A330} \mapsto \mathit{Airbus},\\
&\quad \mathit{A350} \mapsto \mathit{Airbus},\\
&\quad \mathit{A380} \mapsto \mathit{Airbus},\\
&\quad 737 \mapsto \mathit{Boeing},\\
&\quad 747 \mapsto \mathit{Boeing},\\
&\quad \mathit{Superjet}100 \mapsto \mathit{Sukhoi},\\
&\quad C919 \mapsto \mathit{Comac},\\
&\quad \mathit{Global}7500 \mapsto \mathit{Bombardier},\\
&\quad \mathit{Global}8000 \mapsto \mathit{Bombardier},\\
&\quad E170 \mapsto \mathit{Embraer},\\
&\quad E175 \mapsto \mathit{Embraer},\\
&\quad \mathit{Su\_80} \mapsto \mathit{Sukhoi},\\
&\}
\end{aligned}
$$