Concordia University

Department of Computer Science and Software Engineering

SOEN 331 - S

Formal Methods for Software Engineering

Assignment 1: Fundamentals

Witnick-Hans Joseph

ID: 29348743

Alexandra Zana

ID: 40131077

Alexandre Eid

ID: 40155833

October 14, 2021.

Contents

1 General information

Date posted: Thursday 30 September, 2021.

Date due: Thursday, 14 October, 2021, by 23:59.

Weight: 15% of the overall grade.

2 Introduction

You should form a team of **three** members. Each team should designate a leader who will submit the assignment electronically. In case you cannot find a team, please contact me and I will assign you to one. There are **7** problems in this assignment, with a total weight of **100** points. You must prepare all your solutions in LaTeX and produce a single `pdf` file. Name the file after the Concordia id of the person who will submit, e.g. `123456.pdf`.

3 Ground rules

This is an assessment exercise. You may not seek any assistance while expecting to receive credit. **You must work strictly within your team and seek no assistance for this assignment ((e.g. from the teaching assistants, fellow classmates and other teams or external help)**. Please note that you should **not** discuss the assignment during tutorials. I am available to discuss clarifications in case you need any.

All team members are expected to work relatively equally on each problem. The team leader has the responsibility to ensure that the team does not violate this rule. **In your submission, you must include only the names of those team members who contributed to the assignment.** Accommodating someone who did not contribute will result in a penalty.

If there is any problem in the team (such as lack of contribution, etc.), the team leader must contact the instructor as soon as the problem appears.

4 Problems

4.1 Propositional logic (10 pts)

You are shown a set of four cards placed on a table, each of which has a **letter** on one side and a **symbol** on the other side. The visible faces of the cards show the letters **L** and **A**, and the symbols $\square$, and $\diamondsuit$.

Which card(s) must you turn over in order to test the truth of the proposition that *"If a card has a consonant on one side, then it has the symbol $\diamondsuit$ on the other side"*? Explain your reasoning <u>in detail</u> by deciding for <u>each</u> card whether it should be turned over and why. In your answers, apply any and all appropriate validating or non-validating patterns where applicable.

4.1.1 Propositional logic answer

We have two statements:

p: The card has a consonant

q: The card has symbol $\diamondsuit$

$p \to q$, $\neg q \to \neg p$

- Card $\diamondsuit$: Implication does not necessarily entail causation. So even if the card is a $\diamondsuit$ the othe side value could be L or A. It is not the card to turn.

- Card $\square$: Since this card is a vowel, we can check the validity of the contrapositive $\neg q \to \neg p$. This should be a card to turn. This card does not satisfy the second statement that the card is a $\diamondsuit$. So we cannot evaluate the second statement. It is not the card to turn.

- Card L: Since this card is a consonant, we can check the validity of $p \to q$. We can check the back to validate q. This should be a card to turn.

- Card A: This card is like $\neg p$, "A" is not a consonent, and therefore does not help in showing $p \to q$ or $\neg q \to \neg p$

4.2 Predicate logic 1 (10 pts)

In the domain of all people, consider the predicate *disclosed(a, b)* that is interpreted as *"a has disclosed a secret to b."*

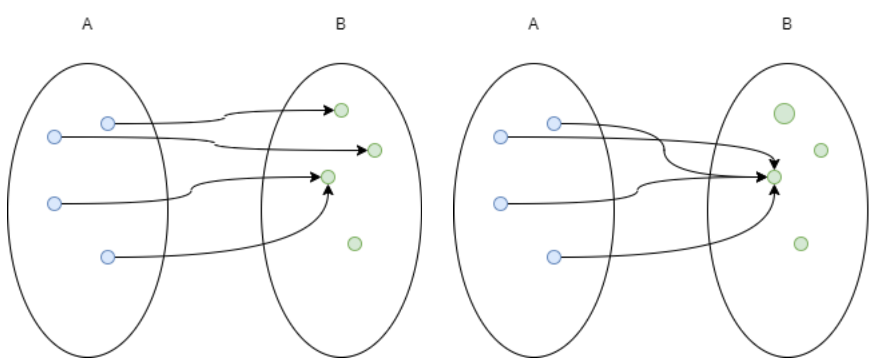
1. How are the following two expressions translated into plain English? Are the two expressions logically equivalent?
   - $\forall a \exists b \ disclosed(a, b)$.
   - $\exists b \forall a \ disclosed(a, b)$.
2. Can we claim that $\forall a \exists b \ disclosed(a, b) \rightarrow \exists b \forall a \ disclosed(a, b)$? Discuss in detail.
3. Can we claim that $\exists b \forall a \ disclosed(a, b) \rightarrow \forall a \exists b \ disclosed(a, b)$? Discuss in detail.

4.2.1 Problem 2 answer

1. translation

   $\forall a \exists b \ disclosed(a, b)$: Everyone has disclosed a secret to someone.

   $\exists b \forall a \ disclosed(a, b)$: There exists someone who has been disclosed a secret by everyone. The two expression are not logically equivalent. Here is a proof by counter example using diagrams. We can see that they are not the same



$\forall a \exists b \ disclosed(a, b)$ $\qquad$ $\exists b \forall a \ disclosed(a, b)$

2. We cannot claim that $\forall a \exists b \ disclosed(a, b) \rightarrow \exists b \forall a \ disclosed(a, b)$.
   In the first statement, we are saying that everyone has disclosed a secret to someone, and thethe other is saying someone has been told be everyone a secret.

The first statement implies that the person b might not be the same person for each person a.

The second statement implies that every a has told a secret to the same b.

3. Yes, we can claim that $\exists\, b\, \forall\, a \;\; disclosed(a, b) \rightarrow \forall\, a\, \exists\, b \;\; disclosed(a, b)$.
   if at least one b has been disclosed a secret by all the a's,
   then we can safely say that all the a have disclosed a secret to a b.

4.3 Predicate logic 2 (10 pts)

Consider the subject “x is a person” and the predicate “x is a mortal”, together with the following list of categorical propositions:

- “No person is immortal.”
- “All people are immortal.”
- “Some people are mortal.”
- “Some people are not mortal.”

1. “Identify each categorical statement with its name (i.e. letter description)”.

   answer:

   ”No person is immortal.”: A Form

   ”All people are immortal.”: E Form

   ”Some people are mortal.”: I Form

   ”Some people are not mortal.”: O Form

2. “Identify universal statements.”

   answer:

   ”No person is immortal.” and ”All people are immortal.”

3. “Identify particular statements.”

   answer:

   ”Some people are mortal.” and ”Some people are not mortal.”

4. “Identify affirmative statements”

   answer:

   ”No person is immortal.” and ”Some people are mortal.”

5. “Identify negative statements.”

   answer:

   ”All people are immortal.” and ”Some people are not mortal.”

6. “Identify statements with opposite truth values”

   answer:

7. “Identify statements that cannot both be true, but could both be false.”

   answer:

   ”No person is immortal.” and ”All people are immortal.”

8. “Identify statements that cannot both be false but could both be true.”

   answer:

   ”All people are immortal.”and ”Some people are mortal.”

9. “Identify pairs of super-subaltern statements.”

   answer:

   ”No person is immortal.” and ”Some people are mortal.”,

   ”All people are immortal.” and ”Some people are not mortal.”

4.4 Ordered structures (10 pts)

Consider a list Λ, deployed to implement a stack Abstract Data Type.

1. Let the head of Λ correspond to the topmost position of the Stack. Implement the body of operations `push(el, Λ)` and `pop(Λ)` (let return element be held in variable topmost) using list construction operations. In both cases a) we assume that appropriate preconditions exist, and b) we can refer to Λ' as the state of the list upon successful termination of one of its operations.

   answer:

   $push(el, \Lambda)\{$
   $\quad \Lambda' = cons(el, \Lambda)$
   $\}$

   $pop(\Lambda)\{$
   $\quad topmost = head(\Lambda)$
   $\quad \Lambda' = tail(\Lambda)$
   $\quad return\ topmost$
   $\}$

2. Let the last element of Λ correspond to the topmost position of the Stack. Implement the body of both operations as above. When applicable, use control flow statements in your answer.

   $push(el, \Lambda)\{$
   $\quad if(head(\Lambda) == null)\{$
   $\qquad \Lambda' = cons(el, \Lambda)$
   $\quad \}else\{$
   $\qquad \Lambda' = concat(\Lambda, list(el))$
   $\quad \}$
   $\}$

4.5 Unordered structures and type declarations (10 pts)

Consider the sets

- Laptop = {Apple,IBM , Sony, HP, Acer , Dell, LG}, and
- Favorite = {Apple, Sony, Dell}.

Answer the following questions:

1. How do we interpret the expression Favorite : $\mathbb{P}$Laptop?

   answer:

   The variable Favorite can assume any value supported by $\mathbb{P}$Laptop (ie: the var Favorite can be any value of the PowerSet of Laptop)

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

   answer:

   Yes; Favorite is of type $\mathbb{P}$Laptop. The powerset Laptop is a legitimate type.

3. What is the nature of the variable in Favorite : $\mathbb{P}$Laptop? (i.e. Atomic or composite? If composite, what type?)

   answer:

   It is a composition. the variabale Favorite: $\mathbb{P}$Laptop is a set of elements, so it is a composition.

4. Is Apple $\in \mathbb{P}$Laptop? Explain why or why no

   answer:

   No; the $\mathbb{P}$Laptop is a list of all possible subsets we can produce from the set Laptop, and the atomic element Apple is not an element of $\mathbb{P}$Laptop.

5. Is {Apple} $\in \mathbb{P}$Laptop? Explain why or why not.

   answer:

   Yes; since {Apple} is a valid element of the $\mathbb{P}$Laptop set. The list element of any atomic element is within the PowerSet of its type.

6. Is $\{\{\}\} \in \mathbb{P}$Laptop?

    answer:

    No, because list Laptop itself does not contain the empty element $\{\}$, it's power set can't contain the element $\{\{\}\}$.

7. Is $\{\} \in \mathbb{P}$Laptop? Explain why or why not.

    answer:

    Yes; since the null/empty element is an element of any power set.

8. If we define variable Favorite : $\mathbb{P}$Laptop, is $\{\}$ a legitimate value for variable Favorite? Explain why or why not.

    answer:

    Yes, because the $\mathbb{P}$Laptop contains the empty list element. Ie, if we remove all the elements contained in Favorite, we will have $\{\}$ set.

9. Is Favorite $\in \mathbb{P}$Laptop? Explain.

    answer:

    Yes; the elements of Favorite are elements in $\mathbb{P}$Laptop, but the variable Favorite itself is not.

10. Is Favorite $\subset \mathbb{P}$Laptop? Explain.

    answer:

    No, Favorite is a variable that can assume any value of $\mathbb{P}$Laptop, but is not a subset of the PowerSet.

4.6 Relational calculus 1 (25 pts)

Consider a system that assigns id's to locations. An id may represent a vehicle in a parking lot, a train in a station, a process in an operating system etc. A location may represent a parking spot, a train station platform, a core in a hardware system etc. The requirements of the system are as follows:

1. Id's are unique.
2. An id is assigned to a single location (and maybe re-assigned subsequently).
3. No multiple id's may be assigned to the same location.

The model of the system is captured by a relation assignment, which is represented as a set as shown below:

$$
\begin{aligned}
&\mathit{assignment} = \\
&\{ \\
&\qquad 001 \mapsto A, \\
&\qquad 002 \mapsto B, \\
&\qquad 003 \mapsto C \\
&\}
\end{aligned}
$$

1. Define the precondition to operation add, that assigns a new id to a location.

   answer:

   id? $\notin$ dom assignment

   location? $\notin$ range assignment

2. Provide two alternative definitions for the main functionality of the operation.

   answer:

   $$\mathit{assignment}' = \mathit{table} \cup \{\mathit{id}? \mapsto \mathit{location}?\}$$

   $$\mathit{assignment} = \mathit{assignment} \oplus \{\mathit{id}? \mapsto \mathit{location}?\}$$

3. What would be the result of calling the operation with id? = 001, and location? = D?

   answer:

$$
\begin{aligned}
&assignment =\\
&\{\\
&\qquad 001 \mapsto D,\\
&\qquad 002 \mapsto B,\\
&\qquad 003 \mapsto C\\
&\}
\end{aligned}
$$

4. Let us get rid of the precondition. What would be the result of calling add with id? = 004, and location? = C? Would you accept or reject the call? If accepted then the pair should be added to the relation. If rejected, it should not be added to the relation.

   answer:

   It would reject the call, as the system does not allow multiple id's to be assigned to the same location

5. Assume that in the absence of a precondition, we attempted to call operation add with some input parameters id? and location?. Under what conditions, if any, would this be acceptable?

   answer:

   This would be acceptable if the id is an existent id.

   In that case we will be overriding and erasing an existing relation.

6. Consider operation reassign that modifies the location of an existing id. Define the precondition to operation reassign.

   answer:

   id? $\notin$ dom assignment

   location? $\notin$ range assignment

7. Define the body of operation reassign.

answer:

$$assignment = assignment \oplus \{id? \mapsto location?\}$$

4.7 Relational calculus 2 (25 pts)

Consider the following relation:

$$cellphones : Model \leftrightarrow Brand$$

where

$$
\begin{aligned}
&cellphones = \\
&\{ \\
&\qquad galaxyA21s \mapsto samsung, \\
&\qquad mate10 \mapsto huawei, \\
&\qquad mate10pro \mapsto huawei, \\
&\qquad galaxyA01 \mapsto samsung, \\
&\qquad iPhone12ProMax \mapsto apple, \\
&\qquad iPhoneSE \mapsto apple, \\
&\qquad galaxyJ2Core \mapsto samsung, \\
&\qquad redmi6A \mapsto xiaomi, \\
&\qquad redmi8 \mapsto xiaomi \\
&\}
\end{aligned}
$$

1. Given the above relation,

   (a) Define the domain of cellphones.

   answer:

   domain cellphones = Model = {galaxyA21s, mate10, mate10pro, galaxyA01, iPhone12ProMax, iPhoneSE, galaxyJ2Core, redmi6A, redmi8}

   (b) Define the range of cellphones.

   answer:

   range cellphones = Brand = {samsung, huawei, apple, xiaomi}

2. Given the following the expression:

   {galaxyA01, galaxyJ2Core, redmi8} $\lhd$ cellphones

   (a) What is the meaning of operator $\lhd$?

   answer: $\lhd$ is a domain restriction, it selects pairs according to the first element.

   (b) What is the value of the expression?

answer:

$\{\text{galaxyA01}, \text{galaxyJ2Core}, \text{redmi8}\} \lhd \text{cellphones} = \{$

$\text{galaxyA01} \mapsto \text{samsung},$

$\text{galaxyJ2Core} \mapsto \text{samsung},$

$\text{redmi8} \mapsto \text{xiaomi}$

$\}$

(c) Where would you deploy the operator $\lhd$ in the context of a database management system?

answer: This operator is used to model database queries.

3. Given the following the expression:

   $\text{cellphones} \rhd \{\text{samsung}, \text{xiaomi}\}$

   (a) What is the meaning of operator $\rhd$?

   answer: $\rhd$ is a range restriction operator; It assignes pairs based on the second element.

   (b) What is the value of the expression? $\text{cellphones} \rhd \{\text{samsung}, \text{xiaomi}\} =$

   $\{$

   $\text{galaxyA21s} \mapsto \text{samsung},$

   $\text{galaxyA01} \mapsto \text{samsung},$

   $\text{galaxyJ2Core} \mapsto \text{samsung},$

   $\text{redmi6A} \mapsto \text{xiaomi},$

   $\text{redmi8} \mapsto \text{xiaomi}$

   (c) Where would you deploy such operator in the context of a database management system?

   answer:

   This operator is used to model database queries.

4. Given the following the expression:

   $\{\text{mate10pro}, \text{iPhoneSE}, \text{galaxyJ2Core}\} \unlhd \text{cellphones}$

   (a) What is the meaning of operator $\unlhd$?

answer:

It is a domain subtraction, it removes all specified element from the domain of the relation.

(b) What is the value of the expression?

answer:

cellphones' = {mate10pro, iphoneSE, galaxyJ2Core} $\lhd$ cellphones = {

galaxyA21s $\mapsto$ samsung,

mate10 $\mapsto$ huawei,

galaxyA01 $\mapsto$ samsung,

iphone12ProMax $\mapsto$ apple,

redmi6A $\mapsto$ xiaomi,

redmi8 $\mapsto$ xiaomi

}

(c) Where would you deploy such operator in the context of a database management system?

answer:

This operator would be deployed when we want to delete elements from the database.

5. Given the following the expression:

cellphones $\rhd$ {apple, xiaomi}

(a) What is the meaning of operator $\rhd$?

answer:

$\rhd$ is a range subtraction operator, it removes all elements from the range of the relation.

(b) What is the value of the expression?

answer:

cellphones' = cellphones $\rhd$ {apple, xiaomi} = {

galaxyA21s $\mapsto$ samsung,

mate10 $\mapsto$ huawei,

mate10pro $\mapsto$ huawei,

galaxyA01 $\mapsto$ samsung,

galaxyJ2Core $\mapsto$ samsung,

}

(c) Where would you deploy such operator in the context of a database management system?

answer:

You would use $\trianglerighteq$ when wanting to modify the content of a database (deleting element, in the case of $\trianglerighteq$).

6. Given the following the expression:

cellphones $\oplus$ {galaxyA51 7→ samsung, mate9 7→ huawei}

(a) What is the meaning of operator $\oplus$?

answer:

$\oplus$ is a relational overide

(b) What is the value of the expression?

answer:

cellphones $\oplus$ {galaxyA51 $\mapsto$ samsung, mate9 $\mapsto$ huawei} = {

galaxyA21s $\mapsto$ samsung,

mate10 $\mapsto$ huawei,

mate10pro $\mapsto$ huawei,

galaxyA01 $\mapsto$ samsung,

iphone12ProMax $\mapsto$ apple,

iphoneSE $\mapsto$ apple,

galaxyJ2Core $\mapsto$ samsung,

redmi6A $\mapsto$ xiaomi,

redmi8 $\mapsto$ xiaomi,

galaxyA51 $\mapsto$ samsung,

mate9 $\mapsto$ huawei,

}

(c) Where would you deploy such operator in the context of a database management system?

answer:

This operator would be used in the event that we want to add or modify elements in the database.

(d) Does the result of the expression have a permanent effect on the database (relation)? If not, describe in detail how would you ensure that such operation would have a permanent effect.

answer:

The expression doesn't have a permanent effect. If we want to have a permanent effect, we would need to have an assignment operation as such:

cellphones' = cellphones $\oplus$ {galaxyA51 $\mapsto$ samsung, mate9 $\mapsto$ huawei}

We need to assign the value of the expression to cellphone to have permanent effect.

5 What to submit

Please submit your `pdf` file at the Electronic Assignment Submission portal

(`https://fis.encs.concordia.ca/eas`)

under **Theory Assignment 1**.