

Concordia University
Department of Computer Science and Software
Engineering

SOEN 331 - S and U
Introduction to Formal Methods
for Software Engineering

Assignment 1

Propositional and Predicate Logic, Structures,
Binary Relations, Functions and Relational Calculus

Team 19

Samuel Boaknin

40009692

Ryan Leyland

40015165

Saleha Tariq

40006997

Meng Susana Ung

40099729

Due Date: February 5, 2021

Contents

1 General information

Date posted: Thursday 28 January, 2021.

Date due: Thursday, February 11, 2021, by 23:59.

Weight: 25% of the overall grade.

2 Introduction

This is a team assignment. Each team should designate a leader who will submit the assignment electronically. There are **9** problems in this assignment, with a total of **100** points. You must prepare all your solutions in \LaTeX and produce a single **pdf** file. Please make sure you include all names and id's of all contributing team members as the authors. Name the file after your team, e.g. `team1.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 project (from the instructor, the teaching assistants, fellow classmates and other teams or external help).** Please note that you should **not** discuss the assignment during tutorials. Failure to do so will result in penalties or no credit.

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. Failure to do so will result in penalties. In your submission, you must include only the names of those people who contributed to the assignment. Accommodating someone who did not contribute will result in penalties.

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 Problem 1 (8 pts)

You are shown a set of four cards placed on a table, each of which has a **number** on one side and a **symbol** on the other side. The visible faces of the cards show the numbers **2** and **7**, and the symbols \square , and \bigcirc .

Which card(s) must you turn over in order to test the truth of the proposition that “*If a card has an odd number on one side, then it has the symbol \square on the other side*”? Explain your reasoning by deciding for each card whether it should be turned over and why.

Solution:

Cards to turn over: 7 and \bigcirc

Let p denote odd number on this side of card and q denote \square on opposite side of card then:

7: check this card due to modus ponens

$$p \rightarrow q, p \therefore q$$

\bigcirc : check this card due to modus tollens

$$p \rightarrow q, \neg q \therefore \neg p$$

2: do not check this card due to inverse error

$$p \rightarrow q, \neg p \therefore \neg q \text{ is incorrect}$$

\square : do not check this card due to converse error

$$p \rightarrow q, q \therefore p \text{ is also incorrect}$$

4.2 Problem 2 (8 pts)

Consider the predicate $asks(a, b)$ that is interpreted as “ a has asked b out on a date.”

1. Translate the following into English: $\forall a \exists b asks(a, b)$ and $\exists b \forall a asks(a, b)$.
2. Can we claim that $\forall a \exists b asks(a, b) \rightarrow \exists b \forall a asks(a, b)$? Discuss in detail.
3. Can we claim that $\exists b \forall a asks(a, b) \rightarrow \forall a \exists b asks(a, b)$? Discuss in detail.

Solution:

1. (a) $\forall a \exists b asks(a, b)$ = Everyone has asked at least one person out on a date.
(b) $\exists b \forall a asks(a, b)$ = There exists a person whom has been asked on a date by everyone.
2. Can we claim that $\forall a \exists b asks(a, b) \rightarrow \exists b \forall a asks(a, b)$? Discuss in detail.

No, modus ponens and modus tollens both fail. There could exist a person whom has not asked the same person out on a date as everyone else. Therefore the LHS would be true and the RHS false which cannot hold for implication.

3. Can we claim that $\exists b \forall a asks(a, b) \rightarrow \forall a \exists b asks(a, b)$? Discuss in detail.

Yes, if a person has been asked on a date by every person then this implies that every person has asked out at least one person on a date.

4.3 Problem 3 (12 pts)

Let $scientist(x)$ denote the statement “x is a scientist”, and $honest(x)$ denote the statement “x is honest.” Formalize the following sentences and indicate their corresponding formal type.

1. “No scientists are honest.”
2. “All scientists are crooked.”
3. “All scientists are honest.”
4. “Some scientists are crooked.”
5. “Some scientists are honest.”
6. “No scientist is crooked.”
7. “Some scientists are not crooked.”
8. “Some scientists are not honest.”

Identify pairs that are contradictories, contraries, subcontraries, and pairs that support subalternation (clearly indicating superaltern and subaltern).

Solution:

The formalization is as follows:

1. $\forall x(scientist(x) \neg \rightarrow honest(x))(TypeE)$
2. $\forall x(scientist(x) \neg \rightarrow honest(x))(TypeE)$
3. $\forall x(scientist(x) \rightarrow honest(x))(TypeA)$
4. $\exists x(scientist(x) \wedge \neg honest(x))(TypeO)$
5. $\exists x(scientist(x) \wedge honest(x))(TypeI)$
6. $\forall x(scientist(x) \rightarrow honest(x))(TypeA)$

7. $\exists x(\textit{scientist}(x) \wedge \textit{honest}(x))(TypeI)$

8. $\exists x(\textit{scientist}(x) \wedge \neg \textit{honest}(x))(TypeO)$

Statements 1 & 2 are contradictory with Statements 5 & 7. Statements 3 & 6 are contradictory with Statements 4 & 8.

Statements 1 & 2 are contrary with Statements 3 & 6. Statements 4 & 8 are subcontrary with Statements 5 & 7.

Statements 1 & 2 are superalterns to the subaltern Statements 4 & 8. Statements 3 & 6 are superalterns to the subaltern Statements 5 & 7.

4.4 Problem 4 (12 pts)

Consider list $\Lambda = \langle w, x, y, z \rangle$, deployed to implement a Queue Abstract Data Type.

1. Let the head of Λ correspond to the front position of the Queue. Implement operations `enqueue(e1, Λ)` and `dequeue(Λ)` using list construction operations. In both cases we can refer to Λ' as the state of the list upon successful termination of one of its operations.
2. Let us now reverse the way we manipulate our data structure and let the head of Λ correspond to the rear of the Queue.
 - (a) What would be the result of `cons(el, Λ)`, and would it be a correct implementation for operation `enqueue(e1, Λ)`?
 - (b) What would be the result of `list(el, Λ)`, and would it be a correct implementation for operation `enqueue(e1, Λ)`?
 - (c) What would be the result of `concat(list(el), Λ)`, and would it be a correct implementation for operation `enqueue(e1, Λ)`?

Solution:

1. (a) `enqueue(e1, Λ)` : $\Lambda' = (\text{append } \Lambda (\text{list } e1))$
(b) `dequeue(Λ)` : let temp = (car Λ); $\Lambda' = (\text{cdr } \Lambda)$; return temp;
2. (a) $\Lambda' = \langle el, w, x, y, z \rangle$, Yes correct as new element added to tail of queue
(b) $\Lambda' = \langle el, \langle w, x, y, z \rangle \rangle$, No 'list' function takes elements as parameters, so now there are only two direct elements inside Λ'
(c) $\Lambda' = \langle el, w, x, y, z \rangle$, Yes correct order, new element added to tail of queue

4.5 Problem 5 (12 pts)

Let $A = \{0, 1, 2, 3, 4\}$ and relations R , S , T , and U on A defined as follows:

$$R = \{(0, 0), (0, 1), (0, 3), (1, 0), (1, 1), (2, 2), (3, 0), (3, 1), (3, 3), (4, 0), (4, 1), (4, 3), (4, 4)\}$$

$$S = \{(0, 1), (1, 1), (2, 3), (2, 4), (3, 0), (3, 4), (4, 0), (4, 1), (4, 4)\}$$

$$T = \{(0, 3), (0, 4), (2, 1), (3, 2), (4, 2), (4, 3)\}$$

$$U = \{(0, 0), (0, 1), (0, 3), (1, 0), (1, 1), (1, 3), (2, 2), (3, 0), (3, 1), (3, 3), (4, 4)\}$$

Fill in the table below, using \checkmark , or \times .

	R	S	T	U
Reflexive	\checkmark	\times	\times	\checkmark
Irreflexive	\times	\times	\checkmark	\times
Symmetric	\times	\times	\times	\checkmark
Asymmetric	\times	\times	\checkmark	\times
Antisymmetric	\times	\checkmark	\checkmark	\times
Transitive	\times	\times	\times	\checkmark
Equivalence	\times	\times	\times	\checkmark
Partial order	\times	\times	\times	\times

4.6 Problem 6 (8 pts)

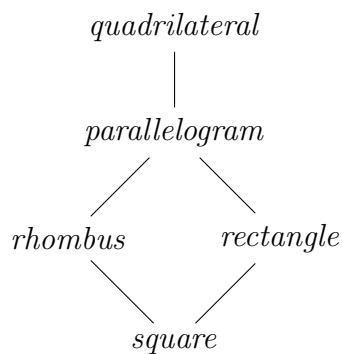
Consider the relation “*is a subtype of*” over the set

$\{\text{rectangle, quadrilateral, square, parallelogram, rhombus}\}.$

1. Is this an *equivalence relation*?
2. Is this relation a *partial order*? If so, create a *Hasse diagram*, and identify *minimal* and *maximal* elements.

Solution:

1. No not an equivalence relation. An equivalence relation must be reflexive, symmetric and transitive. The relation is not symmetric, for example a square is a subtype of a quadrilateral but a quadrilateral is not a subtype of a square as a square is defined to have four right angled corners.
2. Yes, maximal = quadrilateral & minimal = square



4.7 Problem 7 (8 pts)

Consider the set $A = \{w, x, y, z\}$, and the relations

$$S = \{(w, x), (w, y), (x, w), (x, x), (z, x)\}$$

$$T = \{(w, w), (w, y), (x, w), (x, x), (x, z), (y, w), (y, y), (y, z)\}$$

Find the following compositions:

1. $S \circ T$

$$S \circ T = \{(w, w), (w, x), (w, y), (w, z), (x, w), (x, x), (x, y), (x, z), (z, w), (z, x), (z, z)\}$$

2. $T \circ S$

$$T \circ S = \{(w, x), (w, y), (x, w), (x, x), (x, y), (y, x), (y, y)\}$$

3. $T^{-1} \circ S^{-1}$

$$T^{-1} \circ S^{-1} = \{(w, w), (w, x), (w, z), (x, w), (x, x), (x, z), (y, w), (y, x), (z, w), (z, x), (z, z)\}$$

NOTE: Some authors (e.g. Rosen) adopt a different ordering of operands than the one we use in our lecture notes. Please follow the ordering (and the definition) of the lecture notes.

4.8 Problem 8 (12 pts)

Consider sets $A = \{1, 2, 3, 4, 5, 6\}$ and $B = \{a, b, c, d, e, f\}$.

1. Determine the type of the correspondence in each of the following cases, or indicate if the correspondence is not a function.

(a) $\{1 \mapsto b, 2 \mapsto c, 3 \mapsto e, 4 \mapsto d, 5 \mapsto f, 3 \mapsto a\}$

(b) $\{1 \mapsto a, 2 \mapsto d, 3 \mapsto a, 4 \mapsto f, 5 \mapsto d, 6 \mapsto c\}$

(c) $\{1 \mapsto c, 2 \mapsto b, 3 \mapsto d, 4 \mapsto e, 5 \mapsto e, 6 \mapsto f\}$

(d) $\{1 \mapsto b, 2 \mapsto c, 3 \mapsto e, 4 \mapsto d, 5 \mapsto f, 6 \mapsto a\}$

Fill in the table below, using \checkmark , or \times .

	Injective	Surjective	Bijjective	Neither injective nor surjective	Not a function
(a)					
(b)					
(c)					
(d)					

2. Is it possible to construct a function $f : A \rightarrow B$ which is surjective and not injective?
Discuss.

4.9 Problem 9 (20 pts)

Consider the following relation:

$$laptops : Model \leftrightarrow Brand$$

where

$$\begin{aligned} laptops = & \\ & \{ \\ & \quad legion5 \mapsto lenovo, \\ & \quad macbookair \mapsto apple, \\ & \quad xps15 \mapsto dell, \\ & \quad spectre \mapsto hp, \\ & \quad xps13 \mapsto dell, \\ & \quad swift3 \mapsto acer, \\ & \quad macbookpro \mapsto apple, \\ & \quad dragonfly \mapsto hp, \\ & \quad envyx360 \mapsto hp \\ & \} \end{aligned}$$

1. What is the domain and the range of the relation?

$$\begin{aligned} \text{dom } laptops &= \{legion5, macbookair, xps15, spectre, xps13, swift3, macbookpro, \\ &\quad dragonfly, envyx360\} \end{aligned}$$

$$\text{ran } laptops = \{lenovo, apple, dell, hp, acer\}$$

2. What is the result of the expression

$$\{xps15, xps13, swift3, envyx360\} \triangleleft laptops$$

$$\begin{aligned} \{xps15, xps13, swift3, envyx360\} \triangleleft laptops = & \\ & \{ \\ & \quad xps15 \mapsto dell, \\ & \quad xps13 \mapsto dell, \\ & \quad swift3 \mapsto acer, \\ & \quad envyx360 \mapsto hp \\ & \} \end{aligned}$$

What is the meaning of operator \triangleleft and where would you deploy such operator in the context of a database management system?

The \triangleleft operator restricts the domain of the relation and returns the pairs from the relation where the first element matches one of the elements in the domain restrictor. In the context of a database management system, this domain restriction operator could be used to query the database to determine which *Brands* correspond to certain *Models*, given a specific set of *Models*.

3. What is the result of the expression

$$laptops \triangleright \{lenovo, hp\}$$

$$\begin{aligned} laptops \triangleright \{lenovo, hp\} = \\ \{ \\ \quad legion5 \mapsto lenovo, \\ \quad spectre \mapsto hp, \\ \quad dragonfly \mapsto hp, \\ \quad envyx360 \mapsto hp \\ \} \end{aligned}$$

What is the meaning of operator \triangleright and where would you deploy such operator in the context of a database management system?

The \triangleright operator restricts the range of the relation and returns the pairs from the relation where the second element (*Brand*) matches one of the elements in the range restrictor. In the context of a database management system, this range restriction operator could be used to query the database to determine which *Models* correspond to certain *Brands*, given a specific set of *Brands*.

4. What is the result of the expression

$$\{legion5, xps15, xps13, dragonfly\} \triangleleft laptops$$

$$\{ \textit{legion5}, \textit{xps15}, \textit{xps13}, \textit{dragonfly} \} \triangleleft \textit{laptops} =$$

$$\{$$

$$\quad \textit{macbookair} \mapsto \textit{apple},$$

$$\quad \textit{spectre} \mapsto \textit{hp},$$

$$\quad \textit{swift3} \mapsto \textit{acer},$$

$$\quad \textit{macbookpro} \mapsto \textit{apple},$$

$$\quad \textit{envy13} \mapsto \textit{hp}$$

$$\}$$

What is the meaning of operator \triangleleft and where would you deploy such operator in the context of a database management system?

The \triangleleft operator performs a domain subtraction and removes all pairs from the domain of the relation based on their first elements. In the context of a database management system, this operator could be used to remove pairs from the database based on the name of the first element. Here, we removed *laptops* from the domain of the relation whose *Model* matched the ones with the operator.

5. What is the result of the expression

$$\textit{laptops} \triangleright \{ \textit{apple}, \textit{dell}, \textit{hp} \}$$

$$\textit{laptops} \triangleright \{ \textit{apple}, \textit{dell}, \textit{hp} \} =$$

$$\{$$

$$\quad \textit{legion5} \mapsto \textit{lenovo},$$

$$\quad \textit{swift3} \mapsto \textit{acer},$$

$$\}$$

What is the meaning of operator \triangleright and where would you deploy such operator in the context of a database management system?

The \triangleright operator performs a range subtraction and removes all pairs from the codomain of the relation based on their second elements. In the context of a database management system, this operator could be used to remove pairs from the database based on

the name of the second element. Here, we removed *laptops* from the codomain of the relation whose *Brand* matched the ones with the operator.

6. Consider the following expression

$$laptop \oplus \{ideapad \mapsto lenovo\}$$

(a) What is the result of the expression?

$$laptop \oplus \{ideapad \mapsto lenovo\} = \begin{cases} legion5 \mapsto lenovo, \\ macbookair \mapsto apple, \\ xps15 \mapsto dell, \\ spectre \mapsto hp, \\ xps13 \mapsto dell, \\ swift3 \mapsto acer, \\ macbookpro \mapsto apple, \\ dragonfly \mapsto hp, \\ envyx360 \mapsto hp \\ ideapad \mapsto lenovo \end{cases}$$

The relation will have *ideapad* \mapsto *lenovo* appended to it.

(b) What is the meaning of operator \oplus and where would you deploy such operator in the context of a database management system?

The \oplus operator performs a relational override that adds all elements from the RHS of the operator to the relation, if they do not already exist in the relation. In the context of a database management system, this could be used to update the database and add new elements.

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

This result would not have a permanent effect on the relation, it will only return what the relation would look like with the elements added to it. For a permanent effect, you would need to do

$$laptop' = laptop \oplus \{ideapad \mapsto lenovo\}$$

END OF ASSIGNMENT.