NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

QUALIFICATION: BACHELOR OF COMPUTER SCIENCE HONOURS	
QUALIFCATION CODE: 08BCSH	LEVEL: 8
COURSE: FORMAL METHODS	COURSE CODE: FMM810S
SESSION: January 2019	PAPER: Theory
DURATION: 3 Hours	MARKS: 85

SECOND OPPORTUNITY/SUPPLEMENTARY EXAMINATION QUESTION PAPER	
EXAMINER:	Prof. José G. Quenum
MODERATOR:	Dr Gokop Longinus Goteng

This paper consists of 2 pages
(excluding this front page)

INSTRUCTIONS

1. This paper contains 4 questions.
2. Answer all questions on the exam paper.
3. Marks/scores are provided at the right end of each question
4. Do not use or bring into the examination venue books, programmable calculators, mobile devices and other materiaks that may provide you with unfair advantage. Should you be in possession of one right now, draw the attention of the examiner officer or the invigilator.
5. NUST examination rules and regulations apply.

PERMISSIBLE MATERIALS

None

Question 1 .... [25 points]

(a) Give the truth table of the following propositions: [13]

- $(p \land q \Rightarrow r) \land (p \Rightarrow s)$
- $(p \Rightarrow q) \land (\neg q)$

(b) Using the deduction rules, prove the first de Morgan's law: $\frac{\neg (p \lor q)}{\neg p \land \neg q}$ [4]

(c) Using the deduction rules prove the following: $\vdash p \lor \neg p$ [4]

(d) Using the deduction rules prove the following: $p \land q \vdash p \land (q \lor r)$ [4]

Question 2 .... [15 points]

Consider a relation W : $\{\alpha, \gamma, \tau, \pi\} \leftrightarrow \{k, u, m, r\}$ as follows: $\{\alpha \mapsto u, \gamma \mapsto k, \tau \mapsto m\}$

(a) Give in extension the following sets: dom(W) and ran(W) [2]

(b) Give in extension the following sets: $\{\tau\} \lhd$ W and W $\rhd \{r\}$ [4]

(c) Give in extension the following sets: $\{\pi\} \ntriangleleft$ W and W $\ntriangleright \{k, m\}$ [4]

(d) What is the inverse of W? [2]

(e) What is the reflexive closure for W? [3]

Question 3 .... [25 points]

We introduce Danz an abstract data store with a finite number of values (of type V). The state of a Danz object includes:

contents: a sequence of values;

size: the number of values currently in the sequence;

max_size: the contents capacity.

We introduce three (03) operations to manipulate Danz:

initialise to initialise the data store;

insert to insert new values into the contents of the data store;

fetch to retrieve data from the contents of the data store.

Note that after instantiation the capacity of the data store cannot change.

(a) Using the schema notation of the Z specification formalism, propose a specification for Danz [3]

(b) Specify all three operations defined for Danz [8]

(c) In order to provide the user with a proper notification, we introduce a free type, OutCome, which has three values: op_ok, contents_full and contents_empty. Using the new type we can return op_ok when an operation is successful. We can also notify the user with one of the other values in case an insert or a fetch operation fails. Define the free type and using a schema specify the successful notification. [3]

(d) Define the error handling versions of insert and fetch [8]

(e) Define the preconditions for initialise, insert and fetch [3]

Question 4 .... [20 points]

We wish to construct a new type `Terra` with its values bounded by two constants ini_{τ} (lower bound) and fin_{τ} (upper bound). The values in `Terra` should satisfy the following requirements:

- each element in `Terra`, except fin_{τ}, has a successor defined by a function next;
- ini_{τ} cannot be a successor for any value, including itself;
- although fin_{τ} can be a successor for another value, it does not have a successor itself;
- no element should be the successor for more than one element.

Using Z specification formalism, provide an axiomatic definition for `Terra`.