Examination
Mathematical Foundations for Software Engineering

Course codes: DIT022 / DIT023

Date:	2023-08-25
Time:	14:00-18:00
Place:	Lindholmen
Teachers:	Christian Berger Lucas Gren Beatriz Cabrero-Daniel
Visit to e*xam hall:*	Ca. 14:30 and 16:30
Questions:	6
Results:	Will be posted by 2023-09-15.
Grade Limits (DIT022):	Pass (3) 50% Pass with honors (VG) 90%
Grade Limits (DIT023):	Pass (3): 50% Pass with credit (4): 70% Pass with distinction (5): 90%
Allowed aids:	Calculators: Casio FX-82..., Texas TI-30... and Sharp EL-W531... Attached appendix with formulas and tables.

Please observe the following:

- DO NOT write your name on any answer sheet or exam sheet – write the anonymized code instead.
- Write in legible English (unreadable responses mean no points!).
- Motivate your answers and clearly state any assumptions made.
- Start each part of the exam on a new sheet!
- Write only on one side of the paper!
- Only answers written on the answer sheets will be graded, do not write on the exam sheets!
- Before handing in your exam, number and sort the sheets in task order!

NOTE:

Not following these instructions may result in the deduction of points!

Question 1 (5+5+10 = 20pt)
"Logic"

Question 1.1) Answer the following five questions:
(A) When is a compound proposition a contradiction? (1pt)
(B) Is (a ∨ b ∧ ¬a) logically equivalent to TRUE? (1pt)
(C) Can there be an entry of the truth table of a satisfiable compound proposition that is false? (1pt)
(D) Are ((a → b) → a) and (a → (b → a)) logically equivalent? (1pt)
(E) How many rows would the complete truth table for (a → b → c)) have: Hint: you do not need to write the truth table. (1pt)

Please provide your final answers in a clear way, for example:

Q 1.1) a) your answer b) your answer c) your answer d) your answer e) your answer

Question 1.2) Express these statements using predicates and quantifiers and explain their meaning:

- "There is a professor that wants their students to pass the exam." (2pt)
- "No professor wants their students to fail the exam." (3pt)

Question 1.3) Provide the complete truth table (8pt) for the given compound proposition and say to what logical expression is it logically equivalent to (1pt) and why (1pt):
a → ¬ b ∧ (c ∨ b → a) ∨ b

Question 2 (1+2+2 = 5pt)
"Languages and Grammars"

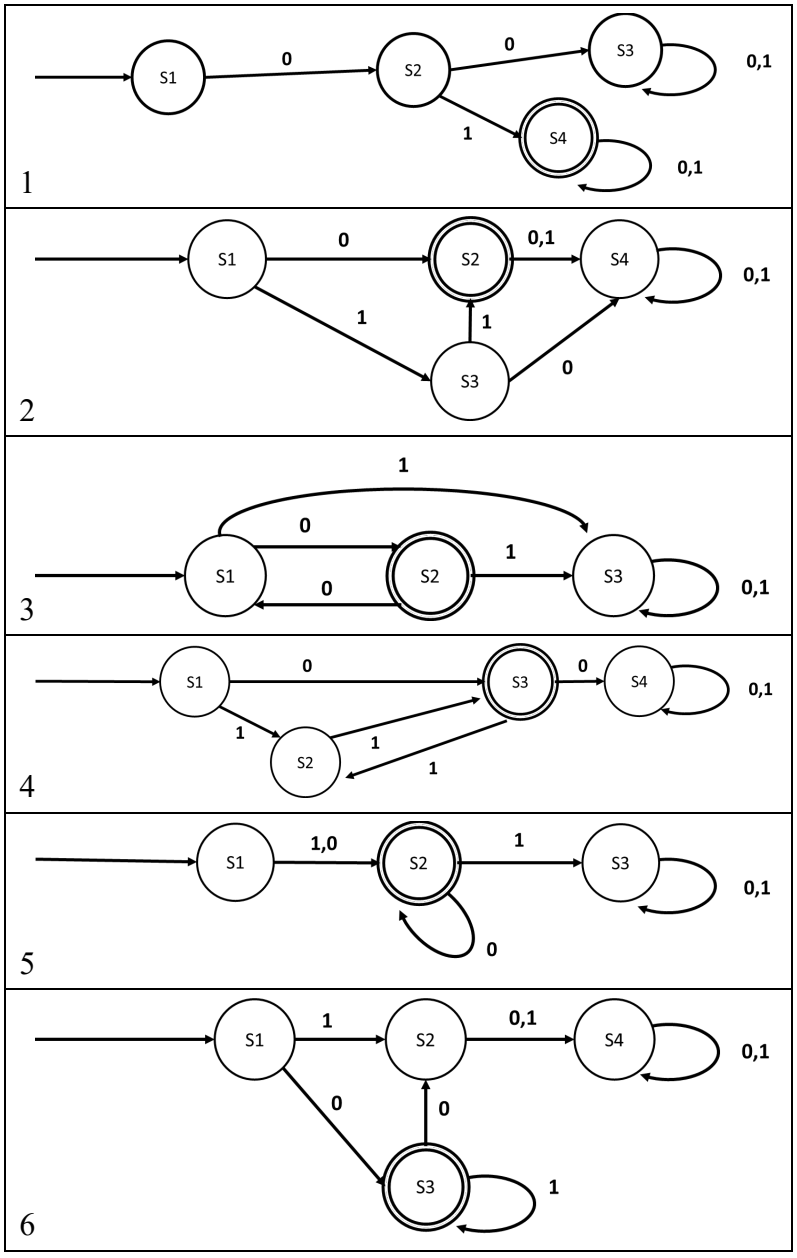
Question 2.1) Let G be the grammar with V = b, e, a, S; T = b, e, a; starting symbol S; and productions S → Sa, S → bSa, and S → e. Give one word that belongs to L(G) that can be generated with this grammar. (1pt)

Question 2.2) Write down the regular expression that denotes all numbers that are a power of 2 in binary notation (with any number of bits). (2pt)
Hint: the answer to the Great Question is 42... or 042.

Question 2.3) Provide two different non-empty words of length 5 that are generated by the Backus-Naur form grammar G and that belong to L(G) (the starting symbol is ⟨code⟩). (2pt)
⟨code⟩ ::= ⟨temp⟩11
⟨temp⟩ ::= ⟨digit⟩11|⟨digit⟩⟨digit⟩1|⟨digit⟩⟨digit⟩⟨digit⟩
⟨digit⟩ ::= 2|3|4|5|7

Question 3 (6+14 = 20pt)
"Automata"

Question 3.1) Connect the following FSM to their outputs. (6pt)
Please provide your final answers in one of the provided blank sheets and in a clear way (e.g., "1A, 2B, 3C, 4D, 5E, 6F").



(A) 0|11
(B) 01*
(C) 0(00)*
(D) (0|11)(11)*
(E) (1|0)0*
(F) 01(0|1)*

Question 3.2) Draw a deterministic, finite state-machine with no output accepting all powers of two numbers (e.g., 0, 2, 4, 8, etc.) in binary notation (with any number of bits). (14pt)

Question 4 (5+5+5+5 = 20pt)
"Complexity and Graphs"

Question 4.1) Trace the execution of the selection sort algorithm over the array below. Note down each pass of the algorithm until the array is sorted. Then, use your trace and match the state using the indexed list below; write down the matching number for every pass of your algorithm (i.e., the state after one complete iteration of the selection sort algorithm). Your answer to this question should be a string of digits (e.g., "12345"). (5pt)
Hint: Write. Numbers. For. Every. Pass.

Input array: [m, t, s, a, l, o]
Indexed list to match your trace against to find the letters for the final answer:
0 [a, l, s, m, t, o]
1 [a, l, m, o, t, s]
2 [o, l, a, s, t, m]
3 [a, t, s, m, l, o]
4 [s, a, o, t, l, m]
5 [o, t, s, a, m, l]
6 [a, l, m, o, s, t]
7 [l, s, m, a, o, t]
8 [a, l, m, s, t, o]
9 [s, m, a, o, t, l]

Question 4.2) What is the complexity of the following code snippet (in terms of Big-O notation)? (5pt)

```
void function1(a, b, c) {
   if (a + b ≤ c) {
      a = a + 1;
      function1(a, b, c);
   }
}
function1(1,2,3);
```

Question 4.3) To establish a big-O relationship, find witnesses (C and k) such that f(x) and g(x) fulfill $|f(x)| \leq C|g(x)|$ whenever $x > k$. Find the least integer n such that f(x) is $O(x^n)$ for $f(x) = (5x^{25})^2 / x^8 + 42$. (5pt)

Question 4.4) There are several properties that algorithms generally share. Match the properties with their definition (5pt):

Properties:
(1) Finiteness
(2) Output
(3) Generality
(4) Correctness
(5) Definiteness

Definitions:
(A) The procedure should be applicable for all problems of the desired form, not just for a set of input values.
(B) The steps of an algorithm must be explained precisely.
(C) From a set of input values an algorithm produces values from a specified set that are the solution to the problem.
(D) An algorithm should produce the desired output after a limited, even if large, number of steps for any input.
(E) An algorithm should produce the correct output values for each set of input values.

Question 5 (5+5 = 10pt)
"Proofs"

Question 5.1) Prove that if x is a positive integer, then 7 divides $3(-18y^2+4y^2)x$. Show your calculations and state which technique you are using to prove it. (5pt)
Hint: the proof might be very direct...

Question 5.2) Is this equality $\sum_{x=0}^{n}(2^x) = 2^{n+1} - 1$ true for all positive integers n? Use mathematical induction to prove it and briefly explain the steps. (5pt)

Question 6 (5+10+10 = 25pt)
"Statistics"

Here's an extract from an interesting paper on maintenance time for software products. You might need to use some of the values reported in the paper (see the tables and figures below) to answer the following exam questions.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 29, NO. 1, JANUARY 2003 77

Predicting Maintenance Performance Using Object-Oriented Design Complexity Metrics

Rajendra K. Bandi, Vijay K. Vaishnavi, *Fellow*, *IEEE*, and Daniel E. Turk, *Member*, *IEEE*

Abstract—The Object-Oriented (OO) paradigm has become increasingly popular in recent years. Researchers agree that, although maintenance may turn out to be easier for OO systems, it is unlikely that the maintenance burden will completely disappear. One approach to controlling software maintenance costs is the utilization of software metrics during the development phase, to help identify potential problem areas. Many new metrics have been proposed for OO systems, but only a few of them have been validated. The purpose of this research is to empirically explore the validation of three existing OO design complexity metrics and, specifically, to assess their ability to predict maintenance time.This research reports the results of validating three metrics, Interaction Level (IL), Interface Size (IS), and Operation Argument Complexity (OAC). A controlled experiment was conducted to investigate the effect of design complexity (as measured by the above metrics) on maintenance time. Each of the three metrics by itself was found to be useful in the experiment in predicting maintenance performance.

Index Terms—Object-oriented metrics, software maintenance, metrics validation, predicting software maintenance time.

Treatment	Mean Maintenance Time		F	P
	Low Complexity	High Complexity		
1	98.02 (s=23.53)	125.09 (s=30.01)	23.41	<0.0001
2	84.30 (s=24.02)	114.58 (s=25.77)	33.16	<0.0001

ANOVAs for Differences by Level of Complexity

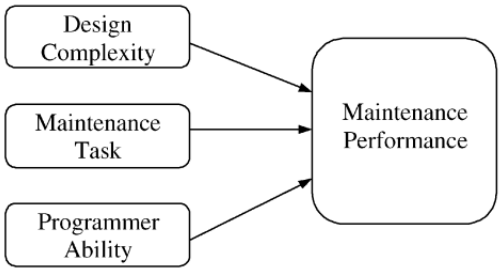
Treatment	Mean Maintenance Time				F	P
	Instructor 1	Instructor 2	Instructor 3	Instructor 4		
1 – Low Compl	98.45 (s=22.04)	99.69 (s=23.16)	96.41 (s=25.98)		0.09	0.91
1 – High Compl		121.30 (s=37.55)		126.60 (s=27.18)	0.22	0.64
2 – Low Compl		88.90 (s=23.24)	85.50 (s=27.47)	81.40 (s=21.54)	0.38	0.68
2 – High Compl	116.30 (s=23.26)	112.44 (s=29.25)			0.20	0.66

ANOVAs for Differences by Instructor

Variable	Test statistic (n = 186)	p-value (α = .05)	β_1	Adjusted R^2
Interaction Level (IL)	5.21	<0.0001	0.04	0.12
Interface Size (IS)	8.00	<0.0001	1.02	0.25
Operation Attribute Complexity (OAC)	7.95	<0.0001	1.45	0.25

Treatment	Mean Maintenance Time		F	P
	Low Complexity	High Complexity		
1 – Spring	97.79 (s=24.56)	121.30 (s=37.55)	5.75	0.02
1 – Summer	98.45 (s=22.04)	126.60 (s=27.18)	14.04	0.0005
2 – Spring	86.56 (s=25.90)	112.44 (s=29.25)	9.77	0.003
2 – Summer	81.40 (s=21.55)	116.30 (s=23.26)	27.17	<0.0001

ANOVAs for Differences by Quarter



Question 6.1) Pick any valid H0 for the statement "there is a relationship between the complexity of a system's design and the maintenance time required to make changes." (5pt)
Select the fully correct solution(s) only. Incorrect answers will invalidate this question (i.e., you will receive 0pt for it). Mark your choice(s), e.g., "6.1) ABC", on your answer sheets, not this exam sheet.

(A) There is a difference in the maintenance time required to make changes to systems, depending on whether design complexity.
(B) There is no relationship between the complexity of a system's design and the maintenance time required to make changes to that system.
(C) There is a nonzero correlation between the complexity of a system's design and the maintenance time required to make changes to that system.
(D) There is no difference in the maintenance time required to make changes to systems, irrespective of whether they have low- or high-complexity designs.
(E) There is not a significant difference in the maintenance time required to make changes to systems, depending on programmer ability.

Question 6.2) Complete these sentences with the words below. *Write your answers clearly (e.g., "6.2,1) ABC, 6.2.2) DEF").* (10pt)

6.2.1) A statistical hypothesis is usually a _____ about a set of parameters of a _____. It is called a hypothesis because it is not known whether it is _____. Now, the objective of a _____ of H0 is not to explicitly determine whether H0 is true but rather to determine if its validity is _____ with the resultant data.

6.2.2) We are often interested in determining whether a particular probabilistic model is appropriate for a given random _____. This determination often reduces to testing whether a given random _____ comes from some specified, or partially specified, _____. Goodness of _____ tests are statistical tests that examine or _____ on if data is following a certain kind of distribution.

A statistical test
B consistent
C population distribution
D true
E probability distribution
F fit
G phenomenon
H sample
I statement
J pass judgement

Question 6.3) Answer the following five questions (10pt):
(1) Based on the test statistics, are the relationship between variable IL (and IS) and maintenance time statistically significant?
(2) Does rejecting the null hypothesis imply a statistically significant difference in maintenance time?
(3) Is the choice of instructor a variable that explains the differences in mean maintenance time?

(4) Is there a correlation between the complexity of a system's design and the maintenance time required to make changes to that system?
(5) Would correlation between the variables (e.g., IL) and mean maintenance time imply causation?