Master in Distributed Software Systems
Entrance Exam

Department of Computer Science

TECHNISCHE UNIVERSITÄT DARMSTADT

August 11, 2016

First Name	
Last Name	
Application Number	
Signature	

Permitted Aids

- None
- Use a black or blue pen with document-proof ink (No green or red color!)

Announcements

- Make sure that your copy of the exam is complete (18 pages).
- Fill out all fields on the front page.
- Put your name and your application number on all pages of this exam.
- The time for solving this exam is 90 minutes.
- All questions of the exam must be answered in English.
- You are not allowed to remove the stapling.
- All tasks can be processed independently and in any order.
- You can use the backsides of the pages if you need more space for your solution.
- Make sure that you have read and understood a task before you start answering it.
- It is not allowed to use your own paper.
- Sign your exam to confirm your details and acknowledge the above announcements.

Topic	1	2	3	4	5	6	7	Total
Points								
Max.	15	15	10	10	15	15	10	90

Name: _____ Application Number: ␣␣␣␣␣␣␣␣␣␣

Topic 1: Automata, Formal Languages and Decidability 15P

a) Context-Free Languages (Cocke, Younger and Kasami; CYK-Algorithm) 7P

Consider the grammar $G1 = (V,E,P,S)$ with $V = \{S,A,B,X,Y\}$, $E = \{a,b\}$, $S = \{S\}$ and $P = \{S \rightarrow AB|CB, D \rightarrow AB|CB, C \rightarrow AD, A \rightarrow a, B \rightarrow b\}$ in Chomsky normal form. Use the CYK-Algorithm to show whether the word $x = aaabbb$ is in $L(G1)$ or not.

b) Context-Free Languages (Pumping Theorem) 4P

Argue with the Pumping Theorem for Context-Free Languages, that the language $L = \{a^m b^m c^m | m \geq 1\}$ is not Context-Free.

Name: _____ Application Number: ⌴⌴⌴⌴⌴⌴⌴⌴⌴⌴

c) Closing Properties for Regular and Context-Free Languages 4P

Complete the following table for closure properties. Enter "Yes" if the language class is closed under the property and "No" if not.

	Union	Intersection	Complement	Concatenation
Regular Languages				
Context-Free Languages				

Name: _____ Application Number: ␣␣␣␣␣␣␣␣␣␣

Topic 2: Software Engineering 15P

Given the following code for comparing two strings

```
1  public class AString {
2    private char value[];
3
4    public boolean equals(AString other) {
5      int n = value.length;
6      if (n == other.getValue().length) {
7         char v1[] = value;
8         char v2[] = other.getValue();
9         for (int i = 0; i < n; i++) {
10             if (v1[i] != v2[i])
11                return false;
12        }
13        return !true;
14     }
15     return false;
16   }
17
18   public char[] getValue() {
19     return value;
20   }
21 }
```

and the following two test cases:

Description	Preinitialized Values	Expected Result
Comparison with longer `AString` (other)	`this.value ="abc"` `other.value = "abct"`	false
Comparison with different `AString` (other)	`this.value ="abc"` `other.value = "act"`	false

(The tasks are found on the following page.)

Name: _____ Application Number: ␣␣␣␣␣␣␣␣␣␣

a) Test Coverage 8P

Analyze (a) the statment coverage and (b) the branch coverage for each of these two test cases. In case of (b) specify for each condition (in line 6,9 and 10) to which value(s) it evaluates.(6P)

If we do not achieve 100% statement and/or branch coverage, specify at most one further test cases to achieve 100% or describe why it is not possible to do so.(2P)

b) Software Process Models 7P

Shortly describe each phase of the waterfall model.(5P)

Shortly describe the biggest problem when using the waterfall model and how agile process try to solve it.(2P)

Name: _____ Application Number: ␣␣␣␣␣␣␣␣

Topic 3: Compiler 10P

Let G be a context-free grammar with: $G = (\{S, A, B, C\}, \{r, s, t, u\}, P, S)$.

The set P contains the following productions in BNF:

```
S  ::=  A r A u B t
A  ::=  ϵ | t
B  ::=  C | r
C  ::=  u | s
```

Complete the following recursive-descent parser for G

```
parseS() {
   parseA();
   if (currentToken == r ) then
      accept()
   else
      error();

   parseA();
   if (currentToken == u) then
      accept()
   else
      error();

   parseB();
   if (currentToken == t) then
      accept()
   else
      error();
}

parseA() { actionTable_A[currentToken](); }
parseB() { actionTable_B[currentToken](); }
parseC() { actionTable_C[currentToken](); }
```

... by filling out the action table below. The action table associates a parsing action (see below) with the next input token, stored by the lexer/scanner in the variable currentToken. Valid actions are:

nop	Do nothing.
accept	Acquire the next token from the lexer and store it in `currentToken`.
`parseA`/`parseB`/`parseC`	Call another parser method.
error	Report error and abort parsing.

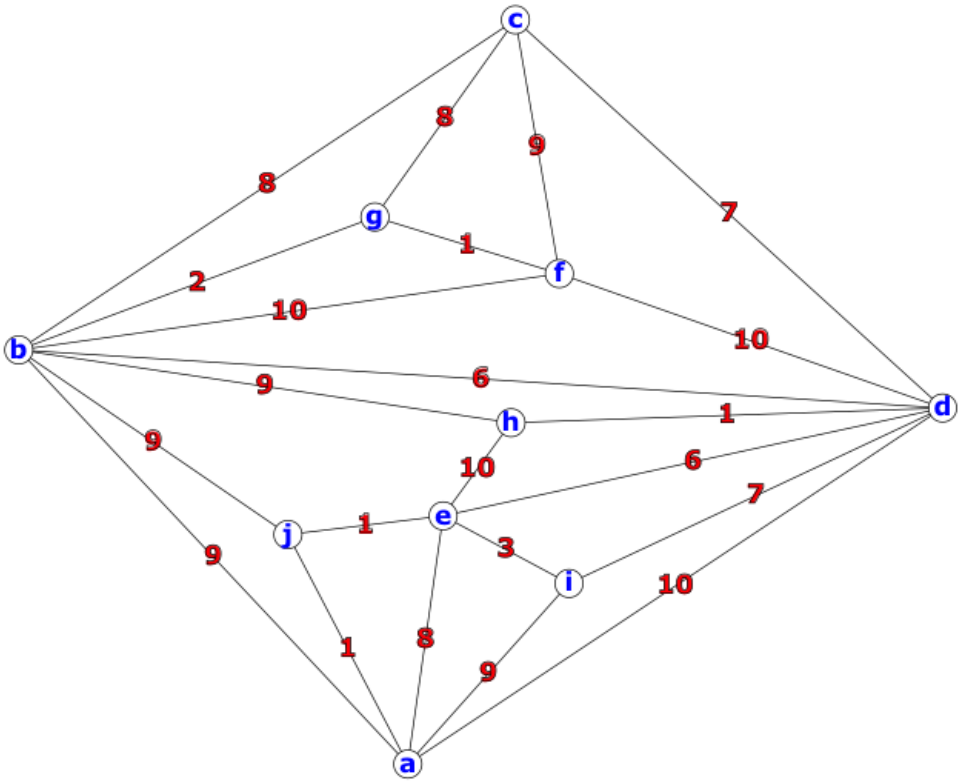
Examples for action table entries are shown for `actionTable_C` in the columns associated with the input tokens `s` and `t`.

curentToken	`actionTable_A`	`actionTable_B`	`actionTable_C`
r			
s			accept
t			error
u			

Name: _____ Application Number: ⌴⌴⌴⌴⌴⌴⌴⌴⌴⌴

Topic 4: Algorithms 10P

As you might know, Dijkstra's algorithm computes the shortest paths from some start node to all other nodes. The algorithm proceeds in rounds, and in each round, one node is finalized. Here is an example network with undirected edges; the red numbers indicate the edge lengths in both directions:



For start node *e*, determine the distance labels of all nodes at the moment when the first two round are finished. Write *INF* for nodes whose distance labels are still infinite:

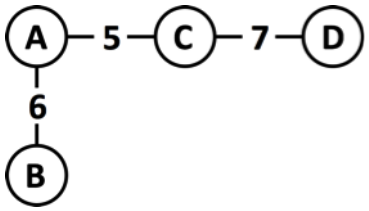
a	b	c	d	e	f	g	h	i	j

Name: _____ Application Number: ⌴⌴⌴⌴⌴⌴⌴⌴⌴⌴

Topic 5: Networks 15P

a) Distance Vector Routing 8P

The network below depicts a router topology. Calculate the initial distance tables for all nodes of the network (before updates are sent). Assume one round, i.e., all nodes received the initial distance vectors from their direct neighbours. Calculate the updated distance tables.



Initial distance tables:

Table A		Via		
		B	C	D
To	B			
	C			
	D			

Table B		Via		
		B	C	D
To	B			
	C			
	D			

Table C		Via		
		B	C	D
To	B			
	C			
	D			

Name: _____ Application Number: ⌴⌴⌴⌴⌴⌴⌴⌴⌴⌴

Table D		Via		
		B	C	D
To	B			
	C			
	D			

After round 1:

Table A		Via		
		B	C	D
To	B			
	C			
	D			

Table B		Via		
		B	C	D
To	B			
	C			
	D			

Name: _____ Application Number: ␣␣␣␣␣␣␣␣␣␣

Table C		Via		
		B	C	D
To	B			
	C			
	D			

Table D		Via		
		B	C	D
To	B			
	C			
	D			

b) DVR Count-to-infinity 2P

Briefly explain the "count-to-infinity problem" in the context of distance vector routing.

c) Multicast 2P

Briefly explain the two terms "broadcast" and "multicast".

d) Transport Layer 2P

Briefly explain the two terms “flow control” and “congestion control”.

e) Queuing 1P

Briefly state the meaning of “Little’s Law” for queuing systems.

Name: _____ Application Number: ⌴⌴⌴⌴⌴⌴⌴⌴⌴⌴

Topic 6: Databases	15P

a) Name two advantages of database normalization.	2P

b) Give a brief definition of the third normal form.	3P

Name: _____ Application Number: ␣␣␣␣␣␣␣␣

c) SQL Queries 5P

Jutta Musterschmidt is the head of a large university library with about 3 million books on computer science, biology, math, and physics. For a financial report, she needs to know the total number and average price of all copies in the computer science part of the library.

Please formulate this query in SQL. Use the following relational schema.

Book

	ISBN	Title	Price
PK	x		
FK			

BookAuthor

	ISBN	ID
PK	x	x
FK	Book.ISBN	Author.ID

Author

	ID	FirstName	LastName
PK	x		
FK			

LibraryPart

	Name	Address	Phone
PK	x		
FK			

Reader

	ID	FirstName	LastName
PK	x		
FK			

Copy

	CopyNr	ISBN	LibraryPart	Reader
PK	x	x		
FK		Book.ISBN	LibraryPart.Name	Reader.ID

Name: _____ Application Number: ␣␣␣␣␣␣␣␣␣␣

d) Transactions 5P

Consider the following two transactions TX1 and TX2 running in parallel against some relational database:

Time	Transaction TX1	Transaction TX2
1	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;	SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
2	UPDATE employees SET salary = salary + 100 WHERE department = "Sales";	
3		UPDATE employees SET salary = salary + 200 WHERE department = "IT";
4	UPDATE employees SET salary = salary + 100 WHERE department = "IT";	
5		UPDATE employees SET salary = salary + 200 WHERE department = "Sales";

Explain briefly, what happens and explain how the database will look like after time step 5. How are such situations typically called?

Name: _____ Application Number: ␣␣␣␣␣␣␣␣␣␣

Topic 7: Object-Oriented Programming 10P

A university consists of a set of members and institutions. In this task, you have to model this topic, with which you are certainly familiar by now. Note: this task assumes a programming language like Java that does not support inheritance from more than one super class.

a) Modeling Using UML 7P

Draw a UML-like diagram that contains the following elements of a university:

- Staff
- Professors
- Assistants
- Students
- The University itself
- A University contains an arbitrary number of Departments.
- Each Department consists of an arbitrary number of Research Groups.

You may add useful super types as you see fit.

Please clearly indicate the connection between types in the diagram. For example, use a dashed line for "has a" relationship and indicate the number of associated elements, * for "arbitrary" or + for "at least one", as shown in the following example for "One car has four wheels".

Note: Clearly indicate abstract classes! You do not have to specify methods or attributes.

Note: If the common super class for all elements is Object, you do not have to state this explicitly. Simply provide independent diagrams in this case.

b) Extending the Model 3P

Apart from departments and research groups, a university also has programmes (in German Studiengang, e.g., Master of Science in Distributed Software Systems). Each student belongs to at least one programme. Programmes belong to one or multiple departments (for example, think of a Joint Bachelor).

First state where you would insert this class into your type hierarchy regarding inheritance and/or association with the other types present, and give a good reason for this decision. You do not have to draw the diagram again!