Name: Matriculation Number:

Final Exam
General CS II (320201)

May 25., 2013

You have two hours(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 0 minutes, leaving you 120 minutes for revising your exam.

You can reach 0 points if you solve all problems. You will only need 100 points for a perfect score, i.e. -100 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

	To be used for grading, do not write here	
prob.	Sum	grade
total	0	
reached		

Please consider the following rules; otherwise you may lose points:

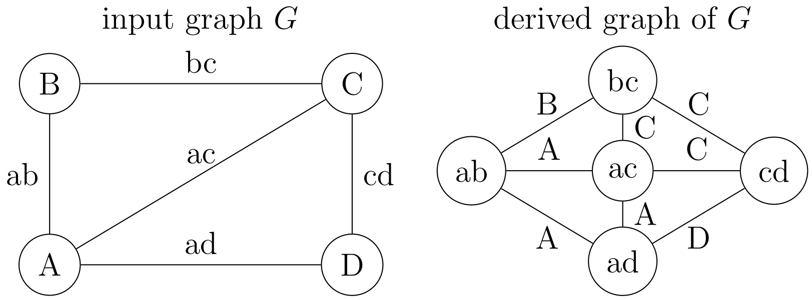
- "Prove or refute" means: If you think that the statement is correct, give a formal proof. If not, give a counter-example that makes it fail.
- Always justify your statements. Unless you are explicitly allowed to, do not just answer "yes" or "no", but instead prove your statement or refer to an appropriate definition or theorem from the lecture.
- If you write program code, give comments!

1 Graphs & Trees

Problem 1.1 (Derived Graph)

10pt

Given a graph G_a we construct its **derived graph** G_b in the following way: For every edge in G_a define a vertex in G_b. If two edges in G_a share a vertex in G_a then their corresponding vertices are connected in G_b. An example is shown below.



1. We call a graph a a **super-graph** if the derived of its derived graph is again a. Prove or refute the existence of graphs that are not super-graphs.

2. Give a necessary and sufficient condition on the edges and vertices that makes a graph a super-graph?

This page was intentionally left blank for extra space

2 Circuits and Positional Number Systems

Problem 2.1 (Mars rover)

Some of you are already familiar with the Mars rover from the Software Engineering lab. Now it's time to implement one of its circuits! The rover has 5 turning states - "hard left turn", "left turn", "straight", "right turn" and "hard right turn". It can receive a command to go left, right or not change the turning state at all. If for example the "left" command is received, the turning state changes one "level" to the left. If it was in a "hard right turn" state, it would go to "right turn". If it was in "straight" state, it will go to "left turn", etc. 15pt

Your task is to design a circuit with 2 bits as input, indicating the received command that can store the current state and update it according to the command. Input of 00 means no change to the turning state, 10 means a "right" command and 11 corresponds to a "left" command. You can assume that the bits will never be 01. The encoding of the states is up to you to determine. You may use D flip-flops and full-adders. All other parts of your circuit must be made out of gates (n-ary gates are allowed). The circuit has no outputs, it just stores and updates the state.

Although not mandatory, writing a short description will help the graders understand your idea and give you partial points.

For your convenience, here's a table you can fill out with your encodings of the states:

hard left	
left	
straight	
right	
hard right	

This page was intentionally left blank for extra space

Problem 2.2 (13-bit TCN)

5pt

1. What is the largest number (in decimal) that can be written using 13-bit two's complement number notation?
2. Convert the obtained decimal number into hexadecimal.
3. What is the smallest number (in decimal) that can be written using 13-bit two's complement number notation?

3 Machine Languages

Problem 3.1 (Longest equal sublist)

Write an ASM program which determines the length of the longest sequence of consecutive equal numbers from a given list of n numbers and the number itself. 15pt

For example, the answer for this list: $1, 1, 2, 5, 2, 2, 2, 6, 6$ is length 3 and number 2.

$D(0)$ contains the number of elements in the list. It is guarantied that the list contains at least one element. $D(10)$ through $D(10 + n - 1)$ contain the elements of the list. At the end of the execution, the length must be saved in $D(1)$ and the number in $D(2)$

Note that you are only allowed to use labels in jump statements. You are very much encouraged to write comments and even a description of your approach, which can help the grader award partial points.

You can find a list of ASM commands (only for reference) on the right.

load i	jump(=) i
store i	jump(>) i
loadi i	jump(<) i
addi i	jump(≥) i
subi i	jump(≤) i
add i	sub i
move S T	jump(≠) i
loadin i j	storein i j
jump i	stop 0

This page was intentionally left blank for extra space

4 Turing Machines

Problem 4.1 (Turing Counter)

Design a Turing Machine that acts as a counter. As input it should take a binary number (of any length) and keep adding 1 to it. The number of bits should never increase and the Turing Machine should keep running forever. It should use the alphabet $\{0, 1, \#\}$. 12pt

Example Input:	##0000##
Output after fist iteration:	##0001##
Output after second iteration:	##0010##
Output after third iteration:	##0011##
and so on.	

This page was intentionally left blank for extra space

5 Internet/WWW/XML

Problem 5.1 (Packets and Protocols)

When you click on a link in your web browser, packets of data are transmitted to the web-server. The server handles those data packets accordingly and sends back another stream of data packets that are in turn received by your browser and interpreted accordingly. Explain this process in detail, highlighting the role of each layer in the protocol stack plays in data transmission. 8pt

Problem 5.2 (Server vs. Client Side Programming)

5pt

1. In the context of web-programming, what is the difference between the server side and the client side programming?
2. Name two typical programming languages.

Problem 5.3 (XML Tree)

Draw a tree structure for the following XML document 4pt

```
<textfile xml:id="foo" xmlns="bar" apple="banana">
  <terminator xml:lang='en'>
    <daftpunk album="RAM" name="touch"/>
  </terminator>
</textfile>
```

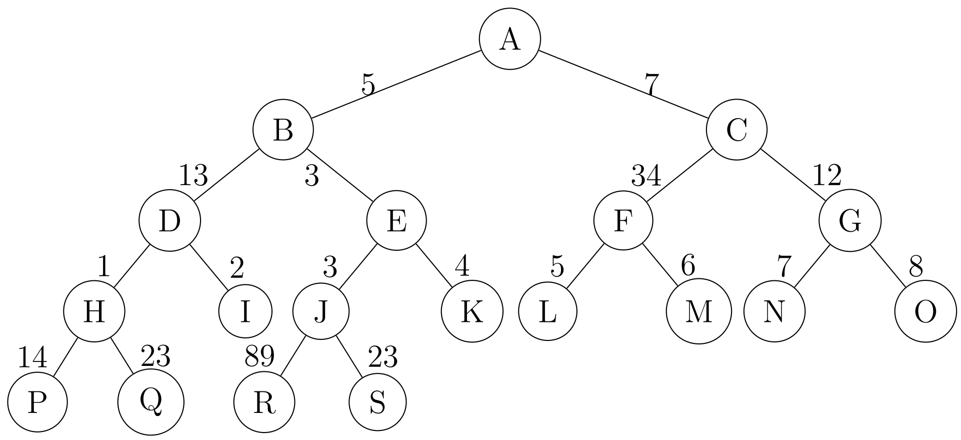
Problem 5.4 (Symmetric Key Algorithm)

Assume you want to write a text file on your computer that only a few can understand. You decide to use symmetric key encryption for this purpose. Design one such algorithm that encrypts the text in your document. Remember there must be a *key*. 5pt

6 Problem Solving and Search

Problem 6.1 (Search Strings)

You are given a graph of the following form: 15pt



None of the nodes in this tree are goal notes. In what order would the nodes be traversed for the following search types:

1. breadth-first search
2. depth-first search
3. uniform-cost search
4. greedy search, where you take the "alphabetic distance" (the number of letters) between the node label and the letter S as the heuristic.

Give the solution as a simple string i.e. ABCDE

This page was intentionally left blank for extra space

7 Programming in Prolog

Problem 7.1 (Mystery Question)

Lets say we have the following Prolog program 12pt

```
bar(A, [], A).
bar([], B, B).
bar([A|RestAs], [B|RestBs], [A|X]) :- A < B, bar(RestAs, [B|RestBs], X).
bar([A|RestAs], [B|RestBs], [B|X]) :- B =< A, bar([A|RestAs], RestBs, X).

apple([], [], []).
apple([A], [A], []).
apple([A,B|Rest], [A|RestA], [B|RestB]) :- apple(Rest, RestA, RestB).

foo([], []).
foo([A], [A]).
foo([A,B|Rest], X) :- apple([A,B|Rest], L1, L2),
                      foo(L1, LX1),
                      foo(L2, LX2),
                      bar(LX1, LX2, X).
```

What is the answer of the following queries:

```
?- bar([1,3,5,7],[2,4,6,8],X).
?- apple([1,2,3,4,5,6,7,8],X,Y).
?- foo([3,1,7,5,4,9,8,2,6],X).
```

Briefly explain what do the functions bar, foo and apple do.

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space