Name: Matriculation Number:

Midterm Exam
General CS II (320102)

March 23, 2015

You have 75 minutes(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 70 minutes, leaving you 5 minutes for revising your exam.

You can reach 76 points if you solve all problems. You will only need 85 points for a perfect score, i.e. -9 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.	1.1	2.1	3.1	4.1	4.2	5.1	6.1	Sum	grade
total	10	10	9	12	15	15	5	76	
reached									

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

- 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, so that we can award you partial credits!
- You may use tags in your $\mathcal{L}$(VM) program to save some (counting) time. Use <string> for tags, where string is a string of lower-case english letters and place the tag before the instruction one would want to jump to when calling `jp` or `cjp`. For a jump place the tag after `jp` and `cjp` and omit writing the relative jump distance.
- Write your program clearly. Should you wish, you may write additional code in a higher level language (HLL) as a comment to help the grader understand what you are trying to do. HLL code without $\mathcal{L}$(VM) code will not give you points.

1 Graphs and Trees

Problem 1.1 (Eulerian paths)

An Eulerian path is a closed path in a connected graph (i.e. there is a path between any two nodes) that starts and ends at the same vertex, while visiting all vertices and traversing all edges *exactly* once. You may know it as the concept behind the "draw the figure without raising your pencil" puzzles. 10pt 10min

Prove that an Eulerian path can be constructed for every connected graph in which all vertices have an even valency.

Hint: A method to generate such a path is sufficient as a proof.

Solution: We start at an arbitrary vertex v and we follow an arbitrary sequence of paths (never repeating an edge) until we reach v again. We will never get stuck because of the even valency, i.e. every time we "enter" a vertex there will be at least one edge that "exits" it. If this arbitrary path goes through all edges then we are done. If not, for each vertex "u" in the path if the vertex has an edge that does not belong to the path we construct an arbitrary path starting and ending at "u" and add it to the Euler path. Therefore, we can always construct an Eulerian path.

2 Combinatorial Circuits

Problem 2.1 (Circuit)

Draw a circuit using only AND, OR and NOT gates that takes a, b and c as inputs and outputs o_1, o_2, o_3, based on the following table. 10pt 10min

a	b	c	o_1	o_2	o_3
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	1	1	1	0

3 Positional Number Systems

Problem 3.1 (Octal Numbers)

Let $A = 1121$ and $B = -50$ be numbers in octal representation (assume the minus behaves similarly to the decimal minus). 9pt 8min

1. Convert the numbers into a corresponding n-bit TCN system. What is the minimal length for converting both A and B?
2. Perform the operations $B + A$ and $B - A$ and convert the results back into the decimal system.

4 Machine Programming

Problem 4.1 (Sum of factorials)

Let $D(0)$ contain a natural number $n > 1$. Write a short assembler program that stores the value of $\sum_{i=1}^{n} i!$ in $D(1)$. You can use the remaining registers as temporary space. 12pt 12min

Note: Assume an assembler operations `MUL` i that changes the value in ACC to $ACC * D(i)$ and `LEQI` i that updates the value in ACC to 1 if $ACC \leq i$ and to 0 otherwise. You may use jump labels.

Solution:

label	instruction	comment
	LOADI 0, STORE 1	$D(1) = 0$ (sum)
	LOADI 1, STORE 2	$D(2) = 1$ (index)
	LOADI 1, STORE 3	$D(3) = 1$ (factorial)
$\langle loop\rangle$	LOAD 0	
	SUB 2, LEQI 0	
	$\texttt{JUMP}_=$ $\langle end\rangle$	
	LOAD 2, ADDI 1, STORE 2	index = index+1
	LOAD 3, MUL 2, STORE 3	factorial = factorial*index
	LOAD 1, ADD 3, STORE 1	sum = sum + factorial
	JUMP $\langle loop\rangle$	
$\langle end\rangle$	STOP 0	

Problem 4.2 (Recursive Programming and Compilation)

15pt
15min

1. Write an μML procedure that given a number n gives returns 1 if the n is prime of form $7k+1$ else returns 0.
2. Then, convert that to $\mathcal{L}(\texttt{VMP})$. Provide comments wherever needed.

Note: You can use labels instead of counters for jumps.

5 Turing Machines

Problem 5.1 (Finding 45)

Given a tape filled with ones and zeroes write a Turing machine that searches the tape for an occurrence of the binary representation of the number 45. If the number 45 is found, the Turing Machine deletes all input and prints a 1. Else, if the number 45 is not found, the turing machine deletes all input and prints a 0. You only need to provide the transition table, and accompanying comments.

15pt
12min

6 Internet

Problem 6.1 (Routers & NICs)

Suppose there are three routers between a source host A and a destination host B. How many network interface controllers does an IP datagram have to pass when sent from A to B?

5pt
3min

Solution: Eight: one each for A and B, and two each for each router (in and out).