Name: Matriculation Number:

Final Exam
General CS I (320101)

December 12., 2014

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

The estimated time for solving this exam is 105 minutes, leaving you 15 minutes for revising your exam.

You can reach 105 points if you solve all problems. You will only need 100 points for a perfect score, i.e. 5 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.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.10	0.11	0.12	Sum	grade
total	12	6	15	4	10	15	5	12	10	6	5	5	105	
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 Mathematical Foundations

Problem 0.1 (Induction for Landau sets)

Recall the definition of the Landau set $O(f)$. Prove the following statement by induction on a: 12pt 12min

$$\forall n, a \in \mathbb{N}. n^a \in n2^n$$

Please note that any other method of proving the statement will earn you ONLY partial credit.

Hint: Do NOT use limits. When using the definition of the Landau sets try to prove that the formula holds for all even n and then prove it for the odd numbers as well.

Solution: This solution is missing some details, but it's complete enough for the general picture.

For $a = 0$ we have $1 \in n2^n$.

For $a = 1$ we have $n \in n2^n$ for obvious reasons, and therefore for all $n \geq N_1$ there are N_1, c_1 with $n \leq c_1 * 2^n$.

Let's assume that for an arbitrary a the statement holds. We then there are N_2, c_2, such that $n^a \leq c_2 * 2^n$ for all $n \geq N_1$.

Multiplying the latter two cases we have:

$$c_1 * c_2 * 2^{2n} \geq n^{a+1} = \frac{2n^{a+1}}{2^{a+1}} \implies \exists c, c * 2^{2n} \geq 2n^{a+1}$$

We proved that all even numbers are in $n2^n$, so we need to prove it for the odd ones as well.

$$2n + 1^{a+1} \leq 2n + 2^{a+1} \leq c * 2^{2n+2} = (2 * c) * 2^{2n+1}$$

Therefore, $2n + 1$ is in $n2^{2n+1}$, and hence(by setting the constant $k = 2 * c$) we have that $\forall n, a \in \mathbb{N}. n^a \in n2^n$

Problem 0.2 Determine whether each of these infinite sets is countable or uncountable. For those that are countably infinite, exhibit a bijective function between the natural numbers and that set. 6pt 6min

1. the integers not divisible by 3
2. the set of all bit strings not containing the bit 0
3. the integers divisible by 5 but not by 7

2 Abstract Data Types and Abstract Procedures

Problem 0.3 (Thesis proposal)

There comes a point during everyone's studies at Jacobs where they have to write a Bachelor's Thesis in order to complete their graduation requirements. CS majors belong in a special subset of majors that are required to write a Thesis Proposal. You are asked to design an ADT for proposals based on the following specifications: 15pt 15min

- A proposal needs a title which is a string, an author(string) and a supervisor(string)
- On the inside it consists of an abstract, an arbitrary number of chapters and a bibliography

- The abstract is a string that explains what the proposal is about
- A chapter has a title and (for the purposes of this problem) a string that represents all the text inside the chapter
- The bibliography is a list of papers, all of which have a title and an author.
- An iteration number that's greater than 0.

Your job is to design an ADT for proposals while having separate sorts for abstract, chapter, bibliography and paper. Feel free to define any additional sorts you may need. You can assume that the sort for strings is already given, so there is no need to redefine them. Write down the paper specified on the right in your ADT

Title	Sth
Author	FS
Supervisor	MK
Abstract	Blabla
Chapters	Introduction, Background and Implementation, all of which contains just the text "bla"
Bibliography	[("First", MK), ("Second", MK)].
Iteration	1

At some point the proposal goes through a number of iterations where the whole text(again for the purposes of this problem) has to be rewritten. Your job is to write an abstract procedure *replace* that takes a proposal, the name of the chapter for which the text is changed and the replacement text and, as you probably guessed by now, replaces the text in that chapter with the replacement text. Define all additional procedures you might need. You can assume that equality on strings is defined.

Problem 0.4 (Equations after Substitution)

Find both triples of terms $\langle A, B, C\rangle$ that satisfy the following equality: $[B/x](g(A,B)) = [C/y](g(f(x),y))$. Show a derivation of your answer. 4pt 4min

Solution: The triples are $\langle f(x), x, x\rangle$ and $\langle x, f(x), f(f(x))\rangle$.

3 Programming in Standard ML

Problem 0.5 (Pascal Triangle)

10pt 10min

Implement an SML function triangle taking an integer as input and generating the Pascal triangle of that depth. The Pascal triangle of depth 4, is given on the right. We do not care about formatting and represent this as [[1],[1,1],[1,2,1],[1,3,3,1]]. That is, the result type should be int list list, where each int list represents a row in the triangle.

```
   1
  1 1
 1 2 1
1 3 3 1
```

Hint: The nth row of the Pascal triangle is defined as [1, x, 1] where x is the list of all sums of 2 consecutive numbers from the $n-1$st row.

Solution:

```
fun mid([1])=[1]
  | mid(h::x::t)= (h+x)::mid(x::t);
(* mid([1,3,3,1]);
  val it = [4,6,4,1] : int list
  adding 1::it gives the next pascal depth *)
```

```
fun pas(1)=[1]
  | pas(n)= 1::mid(pas(n-1));

fun triangle(1)=[[1]]
  | triangle(n)=triangle(n-1)@[pas(n)];
```

Problem 0.6 (Game of Doors)

We play the game of doors like this: 15pt

15min

- We start with 1000 closed doors, numbered from 1 to 1000
- On our first pass, we open every door
- On our second pass, we go to every second door and toggle its state (open $\rightsquigarrow$ closed, closed $\rightsquigarrow$ open)
- On our nth pass, we go to every nth door from the start and repeat the same procedure

After we have passed 1000 times we are curious to find out which doors are open and which are closed. Write an SML function is_open that takes an index of the door and returns whether that door will be open after the 1000th pass.

```
val is_open = fn: int->bool;
-is_open(3);
val it=false: bool
-is_open(4);
val it=true: bool
```

Solution:

```
fun is_open(n) =
    let
        val root = floor(Math.sqrt(real(n)))
    in
     if root*root=n then true else false
    end;
```

---

4 Formal Languages and Codes

Problem 0.7 Given the alphabet $A := \{a, b, c\}$ and a $L := \bigcup_{i=1}^{\infty} L_i$, where $L_1 = \{\epsilon\}$ and L_{i+1} contains the strings x, bbx, xac for all $x \in L_i$. 5pt

5min

1. Is L a formal language?
2. Which of the following strings are in L? Justify your answer

$s_1 = bbac$	$s_2 = bbacc$	$s_3 = bbbac$
$s_4 = acac$	$s_5 = bbbacac$	$s_6 = bbacac$

Solution:

1. L is a formal language as $L_1 \in A^*$ and every step from L_i to L_{i+1} concatenates only elements from A.
2. $s_1, s_4, s_6 \in L$, the rest isn't

5 Boolean Algebra

Problem 0.8 Given the Boolean expression 12pt

$$e := \overline{x_1 + (x_2 * (x_2 + \overline{x_3})) * \overline{x_3}} + (x_3 + x_2) * (x_3 + \overline{x_2})$$

12min

1. Find an equivalent expression of e using only boolean algebra and rules of equivalences on the slides. State explicitly on each step, which equivalence rules you are using. (Truth tables are not allowed.)
2. Generate the truth table for e, determine whether e is satisfiable, valid, falsifiable and/or unsatisfiable.
3. Execute the Quine McClouskey algorithm on f_e to find the minimal cost polynomial.

6 Propositional Logic

Problem 0.9 Recall that $\mathcal{H}^0$ has the two axioms 10pt

1. $K := P \Rightarrow Q \Rightarrow P$
2. $S := (P \Rightarrow Q \Rightarrow R) \Rightarrow (P \Rightarrow Q) \Rightarrow P \Rightarrow R$

10min

and the rules $\dfrac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{A}}{\mathbf{B}}\,\mathrm{MP}$ and $\dfrac{\mathbf{A}}{[\mathbf{B}/X](\mathbf{A})}\,\mathrm{Subst}$

Prove that $(A \Rightarrow B \Rightarrow C) \Rightarrow A \Rightarrow (A \Rightarrow B \Rightarrow C) \Rightarrow A$ in $\mathcal{H}^0$.

Problem 0.10 (Natural Deduction)
Prove the following formula using only the rules of the Natural Deduction calculus. 6pt

$$A \Rightarrow B \Rightarrow C \Rightarrow A \wedge B \wedge C$$

6min

Problem 0.11 (Propositional Tableaux)
Prove the following formula by exhibiting a closed $\mathcal{T}_0$ tableau. 5pt

$$A \wedge B \wedge C \Rightarrow A \Rightarrow B \wedge C$$

5min

7 Intellectual Property and Copyright

Problem 0.12 (CopyLeft)
Briefly state the the copyleft clause in the GNU Public License or in the Creative Commons 5pt

5min

licenses, and explain how it works.

Solution: The copyleft clause states that if a derived work of a licensed work is distributed, then it has to be licensed in exactly the same license as the licensed work.

This makes sure that anybody who wants to make a derived work of the licensed work, they have to decide whether they

- want to distribute it – then they have to license it under the copyleft, and contribute to the Open Source Domain, or
- don't, then they do not have to license it at all (but do not get the benefits of distribution/sale).
