Name: Matriculation Number:

Final Exam
General CS 1 (320101)

December 15, 2009

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

The estimated time for solving this exam is 98 minutes, leaving you 22 minutes for revising your exam.

You can reach 50 points if you solve all problems. You will only need 48 points for a perfect score, i.e. 2 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	2.2	3.1	4.1	5.1	6.1	6.2	6.3	Sum	grade
total	4	8	4	8	6	6	3	6	5	50	
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

4pt

Problem 1.1:

1. Define the concept of an inverse function and discuss the conditions when it exists.
2. For the functions below determine whether they are injective, surjective or bijective. To prove they are not, give counter-examples. Give the inverse function if possible.

$f: \mathbb{N} \to \mathbb{N}$	$x \mapsto 2x$
$h: \{2,3,4\} \to \{8,6,4\}$	$x \mapsto 12 - 2x$
$g: \mathbb{Z} \to \mathbb{N}$	$x \mapsto \|x\|$

where $\mathbb{Z}$ is the set of all integers (positive and negative) and $|x|$ is the absolute value function.

Solution:

- f is injective and not surjective (e.g. $3 \notin f(\mathbb{N})$), so f is not bijective
- h is injective and surjective, thus bijective. Inverse function $6 - y/2$
- Not injective, but surjective

2 Abstract Data Types and Abstract Procedures

8pt

Problem 2.1 (ADT for stack)

1. Design an abstract data type for a stack of natural numbers. The stack should be represented by the actions that constructed it. We have the following operations:

   - push: adds an element to the current stack (usually to the top of the current stack)
   - pop: returns an element from the stack (usually the top most)

Hint: Note that the empty stack and the stack that is constructed by pushing 2 and then popping are different in the envisioned representation, even though they are both empty.

---

2. Once defined, represent the stack 5|1|3|4 in three different ways.

3. Finally, create an abstract procedure ”len” that returns the length of a given stack. For this assume that you are given the procedure *sub* that substracts one from a given natural number. e.g. $sub(s(s(x))) \rightsquigarrow s(x)$.

Solution:

$\mathbb{S}$: stack , $\mathbb{N}$: natural numbers.

$\langle\{\mathbb{N},\mathbb{S}\},\{[o\colon \mathbb{N}],[s\colon \mathbb{N}\to\mathbb{N}],[\text{emptyStack}\colon \mathbb{S}],[\text{pop}\colon \mathbb{S}\to\mathbb{S}],[\text{push}\colon \mathbb{S}\times\mathbb{N}\to\mathbb{S}]\}\rangle$

$5|1|3|4 = \text{push}(\text{push}(\text{push}(\text{push}(\mathit{emptyStack},4),3),1),5)$ $5|1|3|4 = \text{push}(\text{push}(\text{pop}(\text{push}(\text{push}(\text{push}(\text{emptyStac}$

$5|1|3|4 = \text{push}(\text{push}(\text{push}(\text{push}(\text{pop}(\text{push}(\text{emptyStack},7)),4),3),1),5)$

$\langle \mathit{len}{::}\mathbb{S}\to\mathbb{N}\,;\,\{\mathit{len}(\mathit{emptyStack})\rightsquigarrow o, \mathit{len}(\text{push}(l,x))\rightsquigarrow s(\mathit{len}(l)), \mathit{len}(\text{push}(l,x))\rightsquigarrow \mathit{sub}(\mathit{len}(l))\}\rangle$

Problem 2.2 (Substitutions) 4pt

Let $\langle\{\mathbb{A}\}, \{[f\colon \mathbb{A} \times \mathbb{A} \times \mathbb{A} \to \mathbb{A}], [g\colon \mathbb{A} \to \mathbb{A}], [a\colon \mathbb{A}]\}\rangle$ be an abstract datatype and

$$s := f(g(x_{\mathbb{A}}), y_{\mathbb{A}}, f(x_{\mathbb{A}}, h, x_{\mathbb{A}})), \quad t := f(g(g(a)), h, f(g(a), h, g(a)))$$

be two constructor terms of sort $\mathbb{A}$.

1. Find a substitution σ such that $\sigma(s) = t$
2. Compute $\sigma(u)$, where $u := f(g(y_{\mathbb{A}}), z_{\mathbb{A}}, x_{\mathbb{A}})$

Solution:

1. $\sigma := [(g(a))/x_{\mathbb{A}}], [h/y_{\mathbb{A}}]$
2. $f(g(h), z_{\mathbb{A}}, g(a))$

---

3 Programming in Standard ML

8pt

Problem 3.1 (Santa Clause)
Santa left the presents for two brothers under the Christmas tree. Because he didn't have time to divide them, he left you the task to create a program that divides the presents in two sets of the closest value possible. You are given a list of presents described by their value. Your program should output a list containing the total value obtained by each of the two brothers and the presents' distribution.

Example:

```
divide [28, 7, 11, 8, 9, 7, 27];
val it = [([9,11,28],48),([27,7,8,7],49)];
divide [12, 43, 8, 90, 13, 5, 78, 34, 1, 97, 31, 65, 80, 15, 17];
val it = [([17,80,65,97,1,34],294),([15,31,78,5,13,90,8,43,12],295)];
```

Hint: One approach is to create all sublists of presents and choose the best ones.

Hint: You can map the values in the initial list to 1 if they are in the sublist or 0 otherwise.

Solution:

```
fun f(l) = 0::l;
fun g(l) = 1::l;
fun combine 0 = [[]]
| combine(n) = map f(combine(n-1))@ map g(combine(n-1));
fun sum(list) = foldl op+ 0 list;
fun choose_presents([],[], l1, l2) = (l1, l2)
| choose_presents(h1::t1, h2::t2, l1, l2) = if h2 = 0 then
    choose_presents(t1, t2, l1, h1::l2)
else choose_presents(t1, t2, h1::l1, l2)
| choose_presents(_,_, l1, l2) = (l1, l2);
fun best_sol(presents, h::t, L1, L2, best_sum) = let val
(l1, l2) = choose_presents(presents, h, [], [])
in
if sum(l1) <= sum(presents) div 2 andalso sum(l1) > best_sum
then best_sol(presents, t, l1, l2, sum(l1))
else best_sol(presents, t, L1, L2, best_sum)
end
| best_sol(presents, [], L1, L2, best_sum) =
   [(L1, best_sum), (L2, sum(presents) - best_sum)];
fun divide presents = best_sol(presents, combine(length(presents)), [], [], 0);
```

---

4 Formal Languages and Codes

6pt

Problem 4.1: Given the alphabet $A = \{0, \#, @\}$ and a $L := \bigcup_{i=0}^{\infty} L_i$, where

- $L_0 = \{0\}$
- $L_{i+1} = \{xx\#, x@y \mid x, y \in \bigcup_{k=0}^{i} L_k\}$

Write down all the strings in L_2

Solution:

- $L_2 = \{00\#, 00\#00\#\#, 0@00@0\#, 0@0, 0@00\#, 0@0@0, 00\#@0, 00\#@00\#, 00\#@0@0, 0@0@0, 0@0@00\#, 0$

---

5 Boolean Algebra

Problem 5.1: Execute Quine-McCluskey algorithm to get the minimum polynomial for the function with the provided truth table: 6pt

x_1	x_2	x_3	x_4	f
T	T	T	T	T
T	T	T	F	F
T	T	F	T	F
T	T	F	F	T
T	F	T	T	T
T	F	T	F	F
T	F	F	T	F
T	F	F	F	F
F	T	T	T	F
F	T	T	F	T
F	T	F	T	T
F	T	F	F	T
F	F	T	T	F
F	F	T	F	F
F	F	F	T	F
F	F	F	F	T

Solution:

QMC_1 :

x_1	x_2	x_3	x_4
T	T	T	T
T	T	F	F
T	F	T	T
F	T	T	F
F	T	F	T
F	T	F	F
F	F	F	F

x_1	x_2	x_3	x_4
T	X	T	T
X	T	F	F
F	T	X	F
F	T	F	X
F	X	F	F

QMC_2 :

	TTTT	TTFF	TFTT	FTTF	FTFT	FTFF	FFFF
$x_1\, x_3\, x_4$	T	F	T	F	F	F	F
$x_2\, \overline{x_3}\, \overline{x_4}$	F	T	F	F	F	T	F
$\overline{x_1}\, x_2\, \overline{x_4}$	F	F	F	T	F	T	F
$\overline{x_1}\, x_2\, \overline{x_3}$	F	F	F	F	T	T	F
$\overline{x_1}\, \overline{x_3}\, \overline{x_4}$	F	F	F	F	F	T	T

Final result: $f = x_1\, x_3\, x_4 + x_2\, \overline{x_3}\, \overline{x_4} + \overline{x_1}\, x_2\, \overline{x_4} + \overline{x_1}\, x_2\, \overline{x_3} + \overline{x_1}\, \overline{x_3}\, \overline{x_4}$

---

6 Propositional Logic

3pt

Problem 6.1:

1. When do we call a Boolean expression valid?
2. When do we call a Boolean expression satisfiable?
3. Give an example of a Boolean expression that is neither valid nor satisfiable.

Be as formal as possible.

Solution:

- valid in $\mathcal{M}$, iff $\mathcal{M} \models^{\varphi} e$ for all assignments φ.
- satisfiable in $\mathcal{M}$, iff $\mathcal{I}_{\varphi}(e) = \mathsf{T}$ for some assignment φ.
- E.g. $x_1 * \overline{x_1}$.

Problem 6.2 **(Tableau calculus with Santa)** 6pt

1. This year Santas elves were good students and finished studying for GenCS early, so they made up a game. They decide to come up with an expression and use Tableaux to research it. Every elf picks a different variable assignment; and every one whose variable assignment is a model gets a special present from Santa. If we know that 8 elves start the game, how many will be happily holding a present at the end if they start with the expression:

$$A \wedge (B \vee \neg C) \vee (\neg A \vee \neg B \wedge C)$$

2. Oh no! One of the elves was naughty and tried to ruin the game. He mischeviously placed a $\wedge$ instead of $\vee$ in the middle, such that now our elves start with $(A \wedge (B \vee \neg C)) \wedge (\neg A \vee \neg B \wedge C)$. How many elves will now be missing their presents?

Solution:

1.
$$
\begin{array}{c}
A \wedge (B \vee \neg C) \vee (\neg A \vee \neg B \wedge C)^{\mathsf{T}} \\
\begin{array}{c|c}
\begin{array}{c} A \wedge (B \vee \neg C)^{\mathsf{T}} \\ A^{\mathsf{T}} \\ B \vee \neg C^{\mathsf{T}} \\ \begin{array}{c|c} B^{\mathsf{T}} & C^{\mathsf{F}} \end{array} \end{array}
&
\begin{array}{c} \neg A \vee \neg B \wedge C^{\mathsf{T}} \\ \begin{array}{c|c} \begin{array}{c} \neg A^{\mathsf{T}} \\ A^{\mathsf{F}} \end{array} & \begin{array}{c} \neg B \wedge C^{\mathsf{T}} \\ \neg B^{\mathsf{T}} \\ C^{\mathsf{T}} \\ B^{\mathsf{F}} \end{array} \end{array} \end{array}
\end{array}
\end{array}
$$

From the leftmost branch of the four we get models 110 and 111, from the second leftmost we get 110 100, from the second rightmost one (only with AF) we get 000, 001, 010, 011 and from the rightmost branch we get 101 and 001. So in the end all eight possible instantiations: 000, 001, 010, 011, 100, 101, 110, 111 are models and all elves are happy. One can also notice that if we denote $A \wedge (B \vee \neg C)$ by D the second bracket is simply $\neg D$ to argument that the expression is a theorem and is true for any assignment $D \vee \neg D$.

2. Now since by the previous notation we will have an expression of type $D \wedge \neg D$ it will be unsatisfiable and no elf will have a present :((It also works by directly implementing the Tableaux and noticing contradictions everywhere if we start by assigning T to the expreesion just as in I.)

Problem 6.3 (Natural Deduction for PL) 5pt

Given the following inference rules for $\mathcal{ND}^0$:

Introduction	Elimination		Implication
$\dfrac{\mathbf{A} \quad \mathbf{B}}{\mathbf{A} \wedge \mathbf{B}} \wedge I$	$\dfrac{\mathbf{A} \wedge \mathbf{B}}{\mathbf{A}} \wedge E_l$	$\dfrac{\mathbf{A} \wedge \mathbf{B}}{\mathbf{B}} \wedge E_r$	$\dfrac{\begin{array}{c}[\mathbf{A}]^1 \\ \hline\hline \mathbf{B}\end{array}}{\mathbf{A} \Rightarrow \mathbf{B}} \Rightarrow I^1$

Prove that $\mathbf{A} \wedge \mathbf{B} \Rightarrow (\mathbf{B} \Rightarrow \mathbf{A})$. Specify the rules applied at each step.

Solution:

$$\dfrac{\dfrac{\dfrac{\dfrac{[\mathbf{A} \wedge \mathbf{B}]^1}{[\mathbf{B}]^2} \wedge E_r}{\mathbf{A}} \wedge {E_l}^1}{\mathbf{B} \Rightarrow \mathbf{A}} \Rightarrow I^2}{\mathbf{A} \wedge \mathbf{B} \Rightarrow (\mathbf{B} \Rightarrow \mathbf{A})} \Rightarrow I^1$$
