Final Exam
General CS 1 (320101)

December 15, 2008

LAST NAME(s):

FIRST NAME(s):

MATRICULATION NUMBER:

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

You can reach 57 points if you solve all problems. You will only need 49 points for a perfect score, i. e. 8 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 into this box															
prob.	1.1	1.2	2.1	2.2	2.3	3.1	3.2	4.1	4.2	5.1	5.2	6.1	6.2	6.3	Sum	grade
total	3	3	3	2	6	5	8	4	2	5	6	2	3	5	57	
reached																

General Remarks

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

3pt

Problem 1.1: Consider the following statement.
Given two sets A and B. Prove that if there is a subset A' of A which is isomorphic to B, and there is a subset B' of B which is isomorphic to A, then set A and B are isomorphic.

1. Write the statement using MathTalk.
2. Prove or refute it.

Note: Two sets are called isomorphic iff there is a bijection between them. You write $A \sim B$.

3pt

Problem 1.2 (Division by 6)
Prove by induction or refute that for all natural numbers n the following assertion holds: $n(2n^2 - 3n + 1)$ is divisible by 6.

2 Abstract Data Types and Abstract Procedures

3pt

Problem 2.1 (Constructor Terms)
Consider the following abstract data type:

$\mathcal{A} := \langle\{\mathbb{A},\mathbb{B}\},\{[g\colon \mathbb{A}\to\mathbb{B}],[e\colon \mathbb{A}\times\mathbb{B}\to\mathbb{A}],[n\colon \mathbb{B}\to\mathbb{B}],[c\colon \mathbb{A}],[s\colon \mathbb{A}\to\mathbb{A}],[I\colon \mathbb{A}\times\mathbb{A}\to\mathbb{B}]\}\rangle$

Fill in the table below by entering **Y**es or **N**o or / (if not applicable) and give the sort (if term). If an expression is not a term, alter one constructor declaration in order to make the expression a term or explain why this is not possible (use the space below the table).

	expression	term?	ground?	sort	alternative declaration
1	n(g(c))				
2	I(c,g(s(c)))				
3	g(e($x_\mathbb{A}$,g(c)))				

2pt

Problem 2.2 (Applying substitutions)
Given an ADT

$$\langle\{\mathbb{A},\mathbb{B},\mathbb{C}\},\{[a\colon \mathbb{A}],[b\colon \mathbb{B}],[c\colon \mathbb{C}],[f\colon \mathbb{A}\times\mathbb{C}\to\mathbb{B}],[g\colon \mathbb{C}\to\mathbb{A}],[h\colon \mathbb{B}\times\mathbb{B}\to\mathbb{C}]\}\rangle$$

and a term $s = g(h(x_\mathbb{B}, f(y_\mathbb{A}, z_\mathbb{C})))$ check whether each of the following substitutions is valid, apply it if it is or explain why it is not valid.

$\sigma_1 := [b/x_\mathbb{B}],[g(x_\mathbb{C})/y_\mathbb{A}],[h(b,f(a,c))/z_\mathbb{C}]$
$\sigma_2 := [f(a,g(c))/x_\mathbb{B}],[h(b,x_\mathbb{B})/z_\mathbb{C}]$
$\sigma_3 := [f(y_\mathbb{A},c)/x_\mathbb{B}],[g(h(x_\mathbb{B},b))/y_\mathbb{A}]$
$\sigma_4 := [f(y_\mathbb{A},z_\mathbb{C})/x_\mathbb{B}],[b/f(y_\mathbb{A},z_\mathbb{C})]$

Problem 2.3 (Abstract Procedures) 6pt

Given the ADT for binary numbers (think of them as lists of 0 and 1 digits):

$$\langle\{\mathbb{D},\mathbb{B}\},\{[0\colon\mathbb{D}],[1\colon\mathbb{D}],[bit\colon\mathbb{D}\to\mathbb{B}],[addBit\colon\mathbb{B}\times\mathbb{D}\to\mathbb{B}]\}\rangle$$

and the binary operator XOR

XOR	1	0
1	0	1
0	1	0

The bitwise XOR (written as $\oplus$) of two binary numbers performs the logical XOR operation on each pair of corresponding digits (bits). Example:

$$\begin{array}{rr} & 0110 \\ \oplus & 1010 \\ \hline & 1100 \end{array}$$

1. Write down an abstract procedure $\oplus$ that
   - computes the bitwise XOR of two binary numbers if they have equal lengths (leading zeros are accepted)
   - does not terminate otherwise
2. Represent the two numbers 101 and 011 using the ADT above and show the computation process of $\oplus$ on these two numbers.

3 Programming in Standard ML

5pt

Problem 3.1 (Function intersection)

1. Write an SML function that numerically computes all intersections of two math functions and returns a list of the x coordinate of each intersection. Your function should take as input two functions of the form `fn : real -> real` and three more real numbers a,b and ϵ. The result is a list of all values in the interval $[a, b]$ for which the input functions give the same result. The interval should be sampled in steps of size ϵ. In this problem two real numbers are considered equal if their difference is less or equal to 0.001.

2. Using the above write a new function that finds the roots of a math function. The input should be one function of the type `fn : real -> real` and the three additional values defined above. The result should be a list of all points in the interval $[a, b]$ (sampled with a step size of ϵ) for which the input function evaluates to 0.

Example 3.1:

```
fun f (x:real) = x;
fun g (x:real) = 2.0;
intersect f g 0.1 0.0 5.0;
  -> [2.0]
```

Problem 3.2 (Jacobs Path Planner) 8pt

Write an SML function `findMyPath` that given a knowledge base of nodes in the form of directed paths, initial position and the destination, returns a possible route to reach the destination via the nodes. The route returned is simply a list of tuples representing the node connections in the path. In case of no path between the required nodes, raise an exception CantWalkAhead. You do not have to worry about coming up with the optimal solution.

The signature of the function is:

```
(string * string * int) list * string * string -> ((string * string) * int) list
```

Furthermore make two other *SMLlanguage* functions namely:

1. `NetDistance` that takes all the parameters that findMyPath takes and calculates the net distance travelled in the journey returned.

2. `TimeTaken` that takes all the parameters that findMyPath takes plus an additional parameter speed and returns the net time taken for the journey at the given speed. For this part you may use the operator `real` which converts an integer into a real number. e.g.

   ```
   real 4; val it = 4.0 :real;
   ```

Example 3.2:

```
val x = [("Research 1","Research 2",20),("Research 2","Research 3",20),
         ("Research 3","Research 4",20),("Research 2","East Hall",40),
         ("East  Hall","College IV",30)];

findMyPath (x,"Research 1", "College IV");
val it =   [(("Research 1","Research 2"),20),(("Research 2","East Hall"),40),
             (("East Hall","College IV"),30)] : ((string * string) * int) list

findMyPath (x,"Research 1", "RLH");
uncaught exception CantWalkAhead
```

Note: You can use functions from previous parts in subsequent parts without defining them.

This page was intentionally left blank for extra space

4 Formal Languages and Codes

4pt

Problem 4.1 (Codes)

Given the alphabets $P = \{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}, \mathtt{G}, \mathtt{H}\}$ and $Q = \{:,), (, \mathtt{X}\}$ and the function $c\colon P \to Q^+$:

$p \in P$	$c(p)$
A	:)
B	(:
C	:(
D	):
E	:))
F	((:
G	X(
H	:X

1. Is c a character code? (explain or give a counter example)
2. Prove or refute that the extension of c is a string code.
3. Check if c is a prefix code. If not, modify the codewords of c such that it becomes a prefix code.

Problem 4.2 (Greek Codes) 2pt

Consider the follwing character code that maps the Greek letters to symbols from the ASCII table. The Greek letters are represented by their names instead of their symbol:

alpha	beta	gamma	delta	epsilon	zeta	eta	theta
hr	:P	s!	ma	pp	Me	ry	y_
iota	kappa	lambda	mu	nu	xi	omicron	pi
ew	a_	ar	N	st	_	A	!
rho	sigma	tau	upsilon	phi	chi	psi	omega
_Y	_C	He	nd_	e	r	i	Ha

Using the extension of this code encode the string $\zeta\chi\eta\sigma\alpha\psi\nu\delta\gamma$

5 Boolean Algebra

5pt

Problem 5.1 (CNF with Quine-McCluskey)

In class you have learned how to derive the optimal formula for a given function in DNF form using the Quine-McCluskey algorithm. It appears that the same algorithm could be applied to find the optimal formula in CNF form. Think of how this can be done and apply it on the function defined by the following table:

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

Hint: The basic rule used in the QMC algorithm: $a\,x + a\,\overline{x} = a$
also applies for formulas in CNF: $(a + x)\,(a + \overline{x}) = (a)$

Problem 5.2 (Karnaugh-Veitch Diagrams) 6pt

1. Use a KV map to determine <u>all possible minimal</u> polynomials for the function defined by the following truth table:

A	B	C	D	f
F	F	F	F	F
F	F	F	T	T
F	F	T	F	T
F	F	T	T	F
F	T	F	F	T
F	T	F	T	F
F	T	T	F	T
F	T	T	T	T
T	F	F	F	T
T	F	F	T	T
T	F	T	F	F
T	F	T	T	T
T	T	F	F	T
T	T	F	T	T
T	T	T	F	F
T	T	T	T	T

2. How would you use a KV map to find a minimal polynomial for a function with 5 variables? What does your map look like? Which borders in the map are virtually connected? (A simple but clear explanation suffices.)

This page was intentionally left blank for extra space

6 Propositional Logic

2pt

Problem 6.1 (Boolean expressions)

1. What does it mean for an expression e to be falsifiable in a universe $\mathcal{M} := \langle \mathcal{U}, \mathcal{I} \rangle$?
2. If e is falsifiable what can you say about $\neg e$? Justify your answer!

Problem 6.2 (A Hilbert Calculus) 3pt

Consider the Hilbert-style calculus given by the following axioms

1. $P \Rightarrow Q \Rightarrow P$
2. $P \Rightarrow Q \Rightarrow R \Rightarrow P \Rightarrow Q \Rightarrow P \Rightarrow R$

and the rules:

$$\frac{\mathbf{A} \Rightarrow \mathbf{B} \quad \mathbf{A}}{\mathbf{B}}\, MP \qquad \frac{\mathbf{A}}{[\mathbf{B}/X]\mathbf{A}}\, Subst \qquad \frac{\mathbf{A} \Rightarrow \mathbf{B}}{\neg\mathbf{A} \vee \mathbf{B}}\, IMP$$

Prove that $\neg P \vee P$.

Problem 6.3 (Tableau Calculus) 5pt

1. Explain the difference between tableau proof of validity and model generation.
2. Derive a tableau inference rule for $A \Leftrightarrow B^{\mathsf{T}}$. Show the derivation.
3. Generate all models of the following expression: $\neg Q \wedge P \Leftrightarrow Q \wedge \neg P$

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