Final Exam
General CS 1 (320101)

December 13, 2007

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 54 points if you solve all problems. You will only need 47 points for a perfect score, i. e. 7 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	1.3	2.1	2.2	3.1	3.2	4.1	5.1	5.2	6.1	6.2	6.3	Sum	grade
total	3	4	4	4	2	9	4	8	4	3	3	3	3	54	
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
6min

Problem 1.1 (Relations among polynomials)
Prove or refute that $O(n^i) \subseteq O(n^j)$ for $0 \leq i < j, n$ $(i, j, n \in \mathbb{N})$.

4pt
8min

Problem 1.2 (Checking and applying substitutions)
Consider the abstract data type

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

and the term $s := f(g(x_{\mathbb{A}}, h(y_{\mathbb{B}})), z_{\mathbb{C}})$.

For each of the following mappings, check whether they are substitutions. If not, justify that, or otherwise apply the substitution to s.

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

4pt
8min

Problem 1.3 (Are bijective functions with composition a group?)
A group is a set G with a binary operation $*\colon G \times G \to G$, obeying the following axioms:

Closure: G is closed under $*$, i. e. $\forall a, b \in G. a * b \in G$

Associativity: $\forall a, b, c \in G.(a * b) * c = a * (b * c)$

Identity element: $\exists e \in G. \forall a \in G. a * e = e * a = a$.

Inverse elements: $\forall a \in G. \exists a^{-1} \in G. a * a^{-1} = e$.

If, additionally, the following axiom holds, the group is called "commutative" or "Abelian":

Commutativity: $\forall a, b \in G. a * b = b * a$.

- Now prove or refute whether the set of all bijective functions $f : A \to A$ on a set A with the function composition $\circ$ forms a group.
- Is it commutative?

2 Abstract Data Types and Abstract Procedures

Problem 2.1 (Abstract procedure checking sortedness) 4pt 9min

Write an abstract procedure that returns true iff a triple of natural numbers is sorted. We call a triple $\langle a, b, c \rangle$ sorted iff $a \leq b \leq c$. Construct the necessary datatypes for natural numbers and booleans as well.

Problem 2.2 (Constructor terms) 2pt 4min

Consider the following abstract data type:

$$\mathcal{A} := \langle \{\mathbb{A}, \mathbb{B}, \mathbb{T}\}, \{[tuple : \mathbb{A} \times \mathbb{B} \to \mathbb{T}], [first : \mathbb{T} \to \mathbb{A}], [second : \mathbb{T} \to \mathbb{B}], [a : \mathbb{A}], [b : \mathbb{B}], [c : \mathbb{T}]\} \rangle$$

Which of the following expressions are constructor terms (with variables), which ones are ground? Give the sorts for the terms. (No further explanation required! :-)

Answer with **Y**es or **N**o or n/a and give the sort (if term)			
expression	term?	ground?	Sort
$second(tuple(a))$			
$second(tuple(\langle a, b \rangle))$			
$first(tuple(\langle first(x_{\mathbb{T}}), second(c) \rangle))$			
$first(tuple(\langle first(x_{\mathbb{B}}), second(y_{\mathbb{T}}) \rangle))$			

3 Programming in Standard ML

Problem 3.1 (Spammers of the Day!) 9pt 15min

1. Write an SML function `spamfrequency` of type `string list -> (string * int) list`, which takes a list of spam-senders throughout the day and returns a list of the spammers of the day with the number of their spams, for example:

```
spamfrequency(["truthlover", "luv2spam", "deathwish", "earlybird",
"luv2spam", "deathwish", "earlybird", "luv2spam", "earlybird", "luv2spam"]);
→ val it = [("truthlover", 1), ("luv2spam", 4), ("deathwish", 2),
  ("earlybird",3)] : (string * int) list
```

2. Write a function `spamsort` that sorts the list of spammers returned by `spamfrequency` in decreasing order of the number of spams:

```
spamsort([("truthlover", 1), ("luv2spam", 4), ("deathwish", 2),
  ("earlybird", 3)]);
→ val it = [("luv2spam", 4), ("earlybird", 3),
  ("deathwish", 2), ("truthlover", 1)] : (string * int) list
```

3. Write a function `spammerman` that returns the most terrible spammer of the day from a list of spammers as returned by `spamfrequency`:

```
spammerman([("truthlover", 1), ("luv2spam", 4), ("deathwish", 2),
  ("earlybird", 3)]);
→ val it : "luv2spam" : string
```

Note:

1. In parts 2 and 3, you need not handle any exceptions. Just assume that these functions are called with well-formed input, as returned by `spamfrequency`.
2. If there is more than one top-ranked spammer in part 3, just return an arbitrary one of them.

Problem 3.2 (Truth value combinations) 4pt 8min

Write an SML function `combine` of type `int -> int list list` that takes an integer n and returns a list of 2^n lists of length n which represent all combinations of ones and zeros in increasing order (as in a truth table). For example

```
combine 2;
→ val it = [[0,0],[0,1],[1,0],[1,1]] : int list list
```

4 Formal Languages and Codes

Problem 4.1 (Character codes) 8pt 17min

Consider the alphabets

$$A := \{\text{␣}, !, a, \ddot{a}, c, d, e, f, h, i, l, m, n, o, p, r, s, t, u, w, x, y\} \quad (1)$$

$$A^2 := A \times A \quad (2)$$

$$A^3 := A \times A \times A \quad (3)$$

Note: One character of A^3 is a tuple of three characters of A; e. g. $\langle s, u, x\rangle$ would be one such character. For convenience, we write strings over A^3 like *wotsit* nevertheless, if it is unambiguous in the current context that this is actually a string over A^3 (with $\langle f, a, d\rangle$ being the first character).

The space ␣ is just an ordinary character.

---

1. Consider a character code $c_1 \colon A \to A^3$ given by the following table:

␣	a	ä	e	f	i	l	m	n	x	κ
err	ou␣	stm	y␣c	we␣	wis	a␣m	as!	h␣y	hri	κ␣␣

…where κ is a placeholder for any other character in A.

(a) Into what problem would we run without a mapping $\kappa \mapsto$ something being defined?

(b) Would $\kappa \mapsto$ ␣␣␣ work as well?

(c) Does c_1 induce a string code?

2. Compute $c_1'(final␣exäm)$.

3. Consider a function $c_2 \colon c_1(A) \to A^2$ given by the following table:

a␣m	as!	err	h␣y	hri	ou␣	stm	we␣	wis	y␣c	κ␣␣
pp	ar	y␣	a␣	w␣	ha	ye	an	d␣	ne	κ!

c_2 induces a character code on $c_1(A) \to A^2$. Would it also be possible to add some additional mappings in order to make it a character code on $A^3 \to A^2$?

Note: $c_1(A)$ is the image of A under c_1.

---

4. Now consider $c := c_2 \circ c_1$.

(a) Is c a function? If so, what is its type (given as domain $\to$ codomain)?

(b) Is c a character code?

(c) Does c induce a string code?

5. Compute $c'(final␣exäm)$.

5 Boolean Algebra

Problem 5.1 (Quine-McCluskey) 4pt 8min

Use the algorithm of Quine-McCluskey to determine the minimal polynomial of the following function:

$x1$	$x2$	$x3$	f
F	F	F	F
F	F	T	T
F	T	F	F
F	T	T	T
T	F	F	T
T	F	T	T
T	T	F	F
T	T	T	F

Problem 5.2 (CNF with Karnaugh-Veitch Diagrams) 3pt 6min

KV maps can also be used to compute a minimal CNF for a Boolean function. Using the function $f(x1, x2, x3)$ that yields T for $x1^0\,x2^0\,x3^0$, $x1^0\,x2^1\,x3^0$, $x1^0\,x2^1\,x3^1$, $x1^1\,x2^0\,x3^0$, and F for the other inputs, develop an idea (and verify it for this example!) how to do this.

6 Propositional Logic

Problem 6.1: For each of the following boolean expressions, state (and justify!) whether they are satisfiable/falsifiable/unsatisfiable/valid. 3pt 6min

1. $(p \Rightarrow q) \wedge (q \Rightarrow r) \Rightarrow p \Rightarrow r$
2. $x \wedge \neg(x \vee y)$
3. $loves(bill, mary) \wedge loves(mary, bill) \Rightarrow loves(bill, bill)$

Note: You can use whichever method you like: truth tables, boolean algebra, or anything else we introduced in the lecture.

Problem 6.2 (Model generation in Tableau Calculus) 3pt 6min

Find 3 models for the following proposition: $P \vee Q \wedge R \Rightarrow (P \vee Q) \wedge (\neg P \vee R)$

Problem 6.3 (Tableau proof) 3pt 6min

Use the tableaux method to prove the following formula:

1. $\neg P \Rightarrow Q \Rightarrow P \Rightarrow Q \Rightarrow Q$