Name: Matriculation Number:

Final Exam
General CS 1 (320101)

December 13, 2007

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

The estimated time for solving this exam is 107 minutes, leaving you 13 minutes for revising your exam.

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 here														
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															

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

Problem 1.2 (Checking and applying substitutions) 4pt

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. $\sigma_4 := [(f(b, c))/x_{\mathbb{A}}], [(g(f(y_{\mathbb{B}}, c), h(y_{\mathbb{B}})))/y_{\mathbb{B}}]$

Solution:

1. is a substitution
2. is not a substitution; the second item does not map a variable to a constructor term.
3. is not a substitution: it maps a variable of sort $\mathbb{C}$ to a term of sort $\mathbb{A}$
4. is not a substitution: the constructor term is not well-formed.
5. is a substitution

Problem 1.3 (Are bijective functions with composition a group?) 4pt

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

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 $f\colon A \to A$ on a set A with the function composition $\circ$ forms a group.
- Is it commutative?

Solution:

Associativity:

Identity element: the identity function $\lambda x.x$

Inverse elements: f^{-1}, exists due to bijectivity

Commutativity: No, easy counter-example.

---

2 Abstract Data Types and Abstract Procedures

4pt

Problem 2.1 (Abstract procedure checking sortedness)
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.

Solution: Grading: Assuming 4 pt = 100%: 0.5 pt each for every datatype declaration.

Thanks to Ankur Modi and Gordan Ristovski for contributing this exercise.

To do: format!

```
//declare the datatype

<f::N*N->bool, {f(o,o)->true, f(o, s(a))->true, f(s(a), o)->false, f(s(a), s(b))->f(a,b)}>
<fin::N*N*N->bool, {fin(o, a, b)->f(a,b), fin(a, o, b)->false, fin(a, b,
  o)->false, fin(s(a), s(b), s(c))->fin(a, b, c)}>
```

Note: There's a less exciting solution without a helper predicate for binary tuples, here given in SML:

```
datatype nat = o of nat | s of o

fun sorted(z,z,z) = true
  | sorted(z,z,s(_)) = true
  | sorted(z,s(_),z) = false
  | sorted(s(_),z,z) = false
  | sorted(s(_),s(_),z) = false
  | sorted(s(_),z,s(_)) = false
  | sorted(z,s(a),s(b)) = sorted(z,a,b)
  | sorted(s(a),s(b),s(c)) = sorted(a,b,c)
```

Problem 2.2 (Constructor terms) 2pt

Consider the following abstract data type:

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

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
$\text{second}(\mathit{tuple}(a))$			
$\text{second}(\mathit{tuple}(\langle a, b\rangle))$			
$\text{first}(\mathit{tuple}(\langle \text{first}(x_{\mathbb{T}}), \text{second}(c)\rangle))$			
$\text{first}(\mathit{tuple}(\langle \text{first}(x_{\mathbb{B}}), \text{second}(y_{\mathbb{T}})\rangle))$			

Solution:

expression	term?	ground?	Sort
$\text{second}(\mathit{tuple}(a))$	N	/	/
$\text{second}(\mathit{tuple}(\langle a, b\rangle))$	Y	Y	$\mathbb{B}$
$\text{first}(\mathit{tuple}(\langle \text{first}(x_{\mathbb{T}}), \text{second}(c)\rangle))$	Y	N	$\mathbb{A}$
$\text{first}(\mathit{tuple}(\langle \text{first}(x_{\mathbb{B}}), \text{second}(y_{\mathbb{T}})\rangle))$	N	/	/

3 Programming in Standard ML

9pt

Problem 3.1 (Spammers of the Day!)

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)] :
```

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)] :
```

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.

Hint: Make a frequency count on all the spammers and keep track of which spammers have already been checked. Try breaking down the solution into subfunctions for the different tasks.

Solution: Thanks to Ankur Modi for inventing this exercise.
Grading: Assuming 9 pt = 100%, grade 4+3+2.

Problem 3.2 (Truth value combinations) 4pt

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
```

Solution: Grading: Assuming 4 pt = 100%, 1 pt. for the base case(s)

```
fun combine 1 = [[0],[1]]
  | combine n = if (n>0) then
      (map (fn lst => 0::lst) (combine (n-1)))
      @ (map (fn lst => 1::lst) (combine (n-1)))
    else [];
```

---

4 Formal Languages and Codes

8pt

Problem 4.1 (Character codes)
Consider the alphabets

$$\begin{aligned} A &:= \{␣, !, a, ä, c, d, e, f, h, i, l, m, n, o, p, r, s, t, u, w, x, y\} \\ A^2 &:= A \times A \\ A^3 &:= A \times A \times A \end{aligned}$$

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 → codomain)?

   (b) Is c a character code?

   (c) Does c induce a string code?

5. Compute $c'(final␣exäm)$.

Hint: The composition of two functions again is a function. If two functions are in-/sur-/bijective, so is their composition.

Solution: Grading: Assuming 8 pt = 100%, grade 2+1+2+2+1

1. (a) Without a mapping for *all* elements of A defined, c_1 would not be a (total) *function*. Well, in the definition of a code in the lecture it's not explicitly been said whether we're considering partial or total functions, but only total functions make sense for codes. Answers like "we wouldn't be able to encode $s \in A$" are also valid, as they address this problem.

   (b) No, because c_1 would not be injective then.

   (c) Yes, because all code words are equally long and we have a theorem for this case.

2. *we␣wish␣you␣a␣merry␣christmas*!

3. This would not be possible, as $\#(A^3) > \#(A^2)$. By the pigeon hole principle, it is not possible to construct an injective function $A^3 \to A^2$; therefore, it is not possible to construct a character code by definition.

4. Note that the following justifications require some properties of functions to be proven before!

   (a) Yes, because c_1 and c_2 are (total) functions.

   (b) Yes, because c_2 is injective and the composition of injective functions is also injective.

   (c) Yes, because both the extensions of c_1 and c_2 are injective.

5. *and␣a␣happy␣new␣year*

5 Boolean Algebra

4pt

Problem 5.1 (Quine-McCluskey)
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

Solution:

- QMC_1:

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

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

  Thus, the prime implicants are $\overline{x_1}\,x_3$, $\overline{x_2}\,x_3$, $x_1\,\overline{x_3}$.

- QMC_2:

  Prime implicants table:

	FFT	FTT	TFF	TFT
$\overline{x_1}\,x_3$	T	T	F	F
$\overline{x_2}\,x_3$	T	F	F	T
$x_1\,\overline{x_3}$	F	F	T	T

  Thus we can see that the second monomial is not essential. Final result:

$$\overline{x_1}\,x_3 + x_1\,\overline{x_3}$$

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

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.

Hint: Start by grouping F-cells together.

Solution: Grading: Assuming 3 pt = 100%:

- 1 pt for the map
- 0.5 pt for correct grouping
- 1 pt for a reasonable description of the procedure
- 0.5 pt for a correct minimal CNF

6 Propositional Logic

3pt

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

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

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

Solution:

1. valid (and thus also satisfiable)
2. unsatisfiable (use De Morgan), thus also falsifiable
3. satisfiable and falsifiable

Problem 6.2 (Model generation in Tableau Calculus) 3pt

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

Solution: Grading: $\frac{1}{3}$ of the points per model.

Problem 6.3 (Tableau proof) 3pt

Use the tableaux method to prove the following formula:

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

Solution:

$$
\begin{array}{c}
(\neg P \Rightarrow Q) \Rightarrow ((P \Rightarrow Q) \Rightarrow Q)^{\mathsf{F}} \\
\neg P \Rightarrow Q^{\mathsf{T}} \\
(P \Rightarrow Q) \Rightarrow Q^{\mathsf{F}} \\
P \Rightarrow Q^{\mathsf{T}} \\
Q^{\mathsf{F}} \\
P^{\mathsf{F}} \qquad\qquad \left| \; Q^{\mathsf{T}} \right. \\
\begin{array}{c|c|}
\neg P^{\mathsf{F}} & Q^{\mathsf{T}} \\
P^{\mathsf{T}} & \bot \\
\bot &
\end{array} \qquad\qquad \begin{array}{|c} \\ \bot \\ \\ \end{array}
\end{array}
$$
