Name: Matriculation Number:

Midterm Exam
General CS 1 (320101)
October 23. 2006

You have one hour(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 62 minutes, leaving you -2 minutes for revising your exam.

You can reach 61 points if you solve all problems. You will only need 55 points for a perfect score, i.e. 6 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	3.1	3.2	4.1	4.2	Sum	grade
total	4	6	7	8	6	12	6	12	61	
reached										

Good luck to all students who take this test

1 Elementary Discrete Mathematics

4pt

Problem 1.1 (Greek Letters)
Fill in the blanks in the table of Greek letters. Note that capitalized names denote capital Greek letters.

Symbol	Σ	ρ	ξ	δ				
Name					$sigma$	Phi	$omega$	psi

Solution:

Symbol	Σ	ρ	ξ	δ	σ	Φ	ω	ψ
Name	$Sigma$	rho	xi	$delta$	$sigma$	Phi	$omega$	psi

Problem 1.2: Let R and S be (non empty) relations on some given set A. Prove or refute each of the three statements 6pt

1. If R and S are symmetric then $R \cap S$ is symmetric

2. If R is reflexive then all subsets of R are reflexive

3. If R is transitive then R^{-1} is transitive

Solution:

1. Let $\langle x, y \rangle$ be in $R \cap S$. From the definition of intersection $\langle x, y \rangle$ is in R and S. Since R and S are symmetric we get $\langle y, x \rangle \in R$ and $\langle y, x \rangle \in R$ by the definition of symmetry. This implies that $\langle y, x \rangle$ is in $R \cap S$ by the definition of intersection. Hence, $R \cap S$ is symmetric.

2. Let $R := \{\langle a, a \rangle, \langle a, b \rangle, \langle b, b \rangle\}$. R is a reflexive relation on $\{a, b\}$. However the subset $\{\langle a, a \rangle, \langle a, b \rangle\}$ of R is not reflexive.

3. Assume we have $\langle a, b \rangle, \langle b, c \rangle \in R^{-1}$. Then, by definition of the inverse relation we have $\langle b, a \rangle, \langle c, b \rangle \in R$. Since R is transitive, we also know that $\langle c, a \rangle \in R$. This implies again by definition of the inverse relation that $\langle a, c \rangle \in R^{-1}$. Hence R^{-1} is transitive.

Problem 1.3: Using induction, show that the sum of the first n odd numbers, equals n^2, i.e.: 7pt

$$\sum_{i=1}^{n} 2i - 1 = n^2$$

Solution:

Proof: We convince ourselves that for all natural numbers $\sum_{i=1}^{n} 2i - 1 = n^2$

P.1 For the induction we have to consider the following cases:

P.1.1 $n = 1$:

P.1.1.1 then we have $1 = 1^2$ $\square$

P.1.2 $n = 2$:

P.1.2.1 This case is not really necessary, but we do it for the fun of it (and to get more intuition).

P.1.2.2 We have $1 + 3 = 2^2 = 4$ $\square$

P.1.3 $n > 1$:

P.1.3.1 Now, we assume that the assertion is true for a certain $n = k$, i.e. $\sum_{i=1}^{k} (2i - 1) = k^2$.

P.1.3.2 Now, we will try to prove it for $n = k + 1$, i.e. $\sum_{i=1}^{k+1} (2i - 1) = k + 1^2$. We start from writing that: $\sum_{i=1}^{k+1} (2i - 1) = \sum_{i=1}^{k} (2i - 1) + 2(k + 1) - 1 \sum_{i=1}^{k+1} (2i - 1) = k^2 + 2k + 1 = k + 1^2$ $\square$

$\square$

---

2 Substitution

8pt

Problem 2.1: Apply the substitutions $\sigma := [(h(a,f(a),b))/x],[g(c)/y]$ and $\tau := [(f(y))/x],[(g(z))/y],[x/z]$ to the terms $s := g(x,h(y),z)$ and $t := h(g(x,y,g(a,y,x)))$

Hint: Give the 4 result terms $\sigma(s)$, $\sigma(t)$, $\tau(s)$, and $\tau(t)$.

Note: We don't care about the type in this problem, instead we assume that all symbols are appropriately typed.

Solution:

$$\begin{array}{rcl}\sigma(s) & = & g(h(a,f(a),b),h(g(c)),z) \\ \sigma(t) & = & h(g(h(a,f(a),b),g(c),g(a,g(c),h(a,f(a),b)))) \\ \tau(s) & = & g(f(y),h(g(z)),x) \\ \tau(t) & = & h(g(f(y),g(z),g(a,g(z),f(y))))\end{array}$$

3 Abstract Data Types and Abstract Procedures

6pt

Problem 3.1 (SML datatypes vs Abstract Data Types)
Given the SML datatypes

1. `datatype A = a | f of A * A`
2. `datatype B = g of (A * B) -> B`

Write down one abstract data type in math notation representing both SML datatypes at once.

Solution:

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

Problem 3.2 (Mixed Abstract Procedures) 12pt

Consider the following mixed abstract procedures on the abstract data type of natural numbers:

$$\mathcal{F} := \langle f{::}\mathbb{N} \to \mathbb{N}\,;\ \{f(0) \rightsquigarrow 0, f(s(0)) \rightsquigarrow g(s(s(0))), f(s(s(n))) \rightsquigarrow g(s(n))\}\rangle$$

$$\mathcal{G} := \langle g{::}\mathbb{N} \to \mathbb{N}\,;\ \{g(0) \rightsquigarrow 0, g(s(n)) \rightsquigarrow f(n)\}\rangle$$

1. Show the computation process of $f(termappss(s(s(0))))$.
2. Do they terminate on all inputs? Justify your answer!

Solution:

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

---

4 Programming in Standard ML

Problem 4.1 (Call by Value) 6pt

Explain the concept of a "call-by-value" programming language in terms of evaluation order. Give an example program where this affects evaluation and termination, explain it.

Solution: A "call-by-value" programming language is one, where the arguments are all evaluated before the defining equations for the function are applied. As a consequence, an argument that contains a non-terminating call will be evaluated, even if the function ultimately disregards it. For instance, evaluation of the last line does not terminate.

```
fun myif (true,A,_) = A | myif (false,_,B) = B
fun bomb (n) = bomb(n+1)
myif(true,1,bomb(1))
```

Problem 4.2 (Filter and Unique Element) 12pt

Write two functions `filter` and `that` in SML that take a predicate p (a function with result type `bool`) and a list l, where

- `filter` returns the list of all members a of l where $p(a)$ evaluates to `true`.
- `that` returns a if there is exactly one a in l such that $p(a)$ evaluates to `true` and raises the exception `NotUnique` if there are two or more such a, and the exception `NotExistent`, if p evaluates to `false` on all members of l.

Solution:

```
fun filter p (h::t) = if (p h) then h::(filter p t) else (filter p t)
  | filter _ [] = [];

exception NotUnique;
exception NotExistent;

fun that p lst = let val plst = filter p lst
                 in case plst of [] => raise NotExistent
                              | [x] => x
                              | l => raise NotUnique
                 end;
```
