Name: Matriculation Number:

Final Exam
General CS I (320101)

December 13, 2011

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

The estimated time for solving this exam is 0 minutes, leaving you 120 minutes for revising your exam.

You can reach 0 points if you solve all problems. You will only need 48 points for a perfect score, i.e. -48 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.	Sum	grade
total	0	
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

Problem 1.1 (Set properties and induction) 10ptin

Prove the following relations **using induction**:

1. $$(A_1 \cap A_2 \cap \ldots A_n) \cup B = ((A_1 \cup B)) \cap ((A_2 \cup B)) \cap \ldots ((A_n \cup B))$$

2. $$(A_1 \cup A_2 \cup \ldots A_n) \cap B = A_1 \cap B \cup A_2 \cap B \cup \ldots A_n \cap B$$

3. $$(A_1 \backslash B) \cap (A_2 \backslash B) \cap \ldots (A_N \backslash B) = (A_1 \cap A_2 \cap \ldots A_N) \backslash B$$

Hint: You can use the distributivity of intersection over union and of union over intersection. Think whether it also works for set difference.

Hint: Try an induction over the number n of A-sets, whatever these are.

Solution:

1. **Proof**:

   P.1 Base case: $n = 1$
   $A_1 \cup B = A_1 \cup B$ (obvious)

   P.2 Base case: $n = 2$
   $(A_1 \cap A_2) \cup B = (A_1 \cup B) \cap (A_2 \cup B)$ (distributivity of union over intersection)

   P.3 Step case:

$$\begin{aligned} & (A_1 \cap A_2 \cap \ldots A_n \cap A_{n+1}) \cup B \\ = \; & ((A_1 \cap \ldots A_n) \cup B) \cap (A_{n+1} \cup B) \\ = \; & ((A_1 \cup B) \cap (A_2 \cup B) \cap \ldots (A_n \cup B)) \cap (A_{n+1} \cup B) \end{aligned}$$

□

Proof:

2. **P.1** Base case: $n = 1$
   $A_1 \cap B = A_1 \cap B$ (obvious)

   Base case: $n = 2$
   $(A_1 \cup A_2) \cap B = (A_1 \cap B) \cup (A_2 \cap B)$ (distributivity of intersection over union)

   Step case:

$$\begin{aligned} & (A_1 \cup A_2 \cup \ldots A_n \cup A_{n+1}) \cap B \\ = \; & ((A_1 \cup A_2 \cup \ldots A_n) \cup A_{n+1}) \cap B \\ = \; & ((A_1 \cup A_2 \cup \ldots A_n) \cap B) \cup (A_{n+1} \cap B) \\ = \; & (A_1 \cap B) \cup \ldots \cup (A_{n+1} \cap B) \end{aligned}$$

□

P.2 P.3 Proof:

P.1 Base case: $n = 1$

$A_1 \backslash B = A_1 \backslash B$ (obvious)

P.2 Base case: $n = 2$

$(A_1 \backslash B) \cap (A_2 \backslash B) = (A_1 \cap A_2) \backslash B$ (theorem of the elementary set theory)

P.3 Step case:

$$
\begin{aligned}
& (A_1 \backslash B) \cap \ldots \cap (A_n \backslash B) \cap (A_{n+1} \backslash B) \\
= \; & ((A_1 \cap \ldots \cap A_n) \backslash B) \cap (A_{n+1} \backslash B) \\
= \; & ((A_1 \cap A_2 \cap \ldots \cap A_n) \cap A_{n+1}) \backslash B
\end{aligned}
$$

□

Problem 1.2 (Properties of Function Composition) 10pt

Let $f \subseteq A \times B$ and $g \subseteq B \times C$ be functions. Prove or refute the following statements:

1. $g \circ f$ is a function.
2. if f and g are both injective/surjective/bijective, then so is $g \circ f$.
3. $f \circ g$ is also a function and $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.
4. If $f \circ g = \lambda x.x$, then $f = g^{-1}$.

Note: By "refute" we mean "exhibit a counterexample to this claim". Try to make suggestions how the claim can be salvaged.

Solution:

1. To prove this, we have to show that for all given $a \in A$, there is a unique $c \in C$, such that $(g \circ f)(a) = c$. Now, using that f is a function, there is a unique $b \in B$, such that $f(a) = b$, and since g is a function, there is a unique $c \in C$, such that $f(b) = c$. Thus $(g \circ f)(a) = g(f(a)) = g(b) = c$ is unique.

2. To show that $f \circ g$ is injective we choose $(f \circ g)(a) = f(g(a)) = f(g(a')) = (f \circ g)(a')$. As f is injective, we have to have $g(a) = g(a')$ and thus (since g is injective too) $a = a'$, which proves the assertion.

   To show that $f \circ g$ is surjective choose some $c \in C$ and show that it is a pre-image in A. As f is surjective there is a $b \in B$ with $f(b) = c$, and (as g is surjective too), there is an a with $g(a) = b$, so $c = f(g(a)) = (f \circ g)(a)$.

   The the case for bijectivity is proven by combining the two assertions above.

3. $f \circ g$ cannot be a function in general, since $A \neq C$. If $A = C$, then functionhood can be shown just like in case 1.

   The second conjecture is incorrect. First, even we need $A = C$ for the functions to make sense.

   Take for instance $g = \lambda x \in B.c$, where $c \in C$ is arbitrary, then $g \circ f = \lambda x \in A.c$, which is not injective, so it cannot be bijective if $\#(A) \geq 2$. The correct version would be: If $A = C$ and f and g are both bijective, then $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

4. Note that since $\lambda x \in A.x \colon A \to C$, we have $A \subseteq C$. We have to show that for all $a \in A$, $f(a) = g^{-1}(a)$.

---

2 Abstract Data Types and Abstract Procedures

Problem 2.1 (ADT for binary strings) 15pt min

1. Design an ADT to represent binary strings (words over the alphabet $\{0, 1\}$). Give the representation of the binary strings 1100 and 00 in your ADT.
2. Now design an ADT for lists of binary strings.
3. In addition, create an abstract procedure that, given a list of binary strings, sorts it lexicographically according to the ordering of the alphabet $\{0, 1\}$ with $0 < 1$.

Solution:

1. $\langle \{\mathbb{B}\}, \{[1 \colon \mathbb{B}], [0 \colon \mathbb{B}], [put \colon \mathbb{B} \times \mathbb{B} \to \mathbb{B}]\}\rangle$
   $1100 := put(1, put(1, put(0, 0)))$
   $00 := put(0, 0)$

2. ADT for list of words: $\langle \{Lb, \mathbb{B}\}, \{[1 \colon \mathbb{B}], [0 \colon \mathbb{B}], [put \colon \mathbb{B} \times \mathbb{B} \to \mathbb{B}], [nil \colon Lb], [append \colon \mathbb{B} \times Lb \to Lb]\}\rangle$
   The *cmp* procedue compares two binary strings, and returns 1 if the first one is smaller or equal to the second one:

$$\langle cmp{::}\mathbb{B} \times \mathbb{B} \to \mathbb{B}\,;\ \left\{ \begin{array}{l} cmp(0, x) \rightsquigarrow 1 \\ cmp(1, 0) \rightsquigarrow 0 \\ cmp(1, 1) \rightsquigarrow 1 \\ cmp(1, put(0, x)) \rightsquigarrow 0 \\ cmp(1, put(1, x)) \rightsquigarrow 1 \\ cmp(put(0, x), put(0, y)) \rightsquigarrow cmp(x, y) \\ cmp(put(0, x), put(1, y)) \rightsquigarrow 1 \\ cmp(put(1, x), put(1, y)) \rightsquigarrow cmp(x, y) \\ cmp(put(1, x), put(0, y)) \rightsquigarrow 0 \end{array} \right\} \rangle$$

   $\langle if{::}\mathbb{B} \times (Lb \times Lb) \to Lb\,;\ \{if(0, x, y) \rightsquigarrow y, if(1, x, y) \rightsquigarrow x\}\rangle$

   (below m stands for *merge*, and a - for *append*)

$$\langle merge{::}Lb \times Lb \to Lb\,;\ \left\{ \begin{array}{l} merge(nil, x) \rightsquigarrow x \\ merge(x, nil) \rightsquigarrow x \\ m(a(x, xs), a(y, ys)) \rightsquigarrow if(cmp(x, y), a(x, m(xs, a(y, ys))), a(y, m(a(x, xs), ys))) \end{array} \right\}$$

   The *split* procedure started with tsecond parameter 0 returns the binary strings at odd positions, when started with second parameter 1 - gives the binary strings at even positions.
   $\langle split{::}Lb \times \mathbb{B} \to Lb\,;\ \{split(nil, x) \rightsquigarrow nil, split(append(x, xs), 0) \rightsquigarrow append(x, split(xs, 1)), split(append(x,$
   $\langle sort{::}Lb \to Lb\,;\ \{sort(nil) \rightsquigarrow nil, sort(appendx, nil) \rightsquigarrow append(x, nil), sort(x) \rightsquigarrow merge(sort(split(x, 0)), so$

Problem 2.2 (Substitutions) 5pt min

Given the ADT $\langle\{\mathbb{A}\}, \{[a\colon \mathbb{A}], [b\colon \mathbb{A}], [f\colon \mathbb{A} \to \mathbb{A}], [g\colon \mathbb{A} \times \mathbb{A} \to \mathbb{A}], [h\colon \mathbb{A} \times (\mathbb{A} \times \mathbb{A}) \to \mathbb{A}]\}\rangle$ and the following constructor terms of sort $\mathbb{A}$:

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

your tasks are:

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

Solution:

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

---

3 Programming in Standard ML

Problem 3.1 (Mutual Recursion) 10pt

1. Implement the following functions in SML. Do not forget to raise exceptions when needed.

   (a)
$$f(x) = \begin{cases} 5 \cdot g(x-1) & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$
$$g(x) = \begin{cases} f(x) + 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases}$$

   (b) Functions even and odd that determine whether the input x is even or odd.

   (c)
$$h(x,y) = \begin{cases} a(x) \cdot b(y) & \text{if x is even }, y > 0 \\ a(x) - b(y) & \text{if x is odd }, y > 0 \\ c(x+1) & x = 0 \end{cases}$$
$$a(x) = \begin{cases} 1 & \text{if } x = 0 \\ h(x \text{ div } 2, x \text{ div } 2) & \text{if } x > 0 \end{cases}$$
$$b(x) = \begin{cases} 33 & x = 0 \\ h(x \text{ mod } 2, x \text{ div } 2) & \text{if } x > 0 \end{cases}$$
$$c(x) = \begin{cases} h(x \text{ div } 2, x \text{ mod } 2) & \text{if } x > 0 \\ x + 3 & \text{if } x = 0 \end{cases}$$

2. What do the functions $f(x)$ and $g(x)$ compute?

3. Does the function $a(x)$ terminate for all inputs?

Solution:

1.
```
exception negative;
fun f(0) = 0
  | f(n) = if n<0 then raise negative else 5*g(n-1)
and g(0) = 0
  | g(n) = if n<0 then raise negative else 1 + f(n);

fun even(0) = true
  | even(1) = false
  | even(n) = odd(n-1)
and odd(1) = true
```

```
  | odd(0) = false
  | odd(n) = even(n-1);

fun h(0,y) = c(0+1)
  | h(x,y) = if y<= 0 then raise negative else if x<0 then raise negative
else if even(x) then a(x)*b(x) else a(x) - b(x)
and a(0) = 1
  | a(x) = if x<0 then raise negative else h(x div 2, x div 2)
and b(0) = 33
  | b(x) = if x<0 then raise negative else h(x mod 2, x div 2)
and c(0) = 3
  | c(x) = if x<0 then raise negative else h(x div 2, x mod 2);
```

2. $f(x) = 5 \cdot g(x-1) = 5 \cdot (f(x-1)+1) = 25 \cdot (f(x-2)+1) + 5 = \cdots = 5^x + 5^{x-1} + \cdots + 5 = 5 \cdot \frac{5^x-1}{5-1} = 5 \cdot \frac{5^x-1}{4}$ $g(x) = 1 + f(x) = 1 + 5 \cdot \frac{5^x-1}{4}$ if $x < 0$.

3. $h(0, y)$ would not terminate, since $h(0, y) = c(1) = h(1 mod 2, 1) = h(0, 1) = c(1) = \ldots$

Problem 3.2 (Partitions and Sums) 20pt

1. Design an SML function that takes a list L and returns a list containing all the sublists of L (i.e. the power set of L interpreted as a set). Signature and example:

```
val powerSet = fn : 'a list −> 'a list list
− powerSet [1,2,3];
val it = [[1,2,3],[1,2],[1,3],[1],[2,3],[2],[3],[]] : int list list
```

2. Now design an SML function which takes as argument a list L containing only positive, distinct integers. The function returns the largest element in L which can be written as the sum of some other (distinct) elements in L. If no such number is found, return 0. Signature and example:

```
val largest = fn : int list −> int
 − largest [3,1,15,7,5,40];
val it = 15 : int
```

Explanation: 15 is the largest number in the list which can be written as a sum of some other distinct numbers in the list: $3 + 7 + 5$.

Hint: You can use the powerSet function that you defined under **??**.

Solution:

```
Control.Print.printLength := 1000;

fun append x ll = map (fn ls => x :: ls) ll

fun powerSet [] = [[]]
  | powerSet (h :: t) = let val ps = powerSet t in append h ps @ ps end

fun find x [] = false
  | find x (h :: t) = if x = h then true else find x t

fun sum ls = foldl op+ 0 ls

fun getMax [] ls max = max
  | getMax (h :: t) ls max =
      let val s = sum h
      in if find s ls andalso not (find s h) andalso s > max
           then getMax t ls s
           else getMax t ls max
      end

fun largest ls = getMax (powerSet ls) ls 0
```