Name: Matriculation Number:

Midterm Exam
General CS I (320101)

October 27, 2014

You have 75 minutes(sharp) for the test;
Write the solutions to the sheet.

The estimated time for solving this exam is 0 minutes, leaving you 75 minutes for revising your exam.

You can reach 0 points if you solve all problems. You will only need 50 points for a perfect score, i.e. -50 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:

- 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, so that we can award you partial credits!

1 GenCS Classics and Induction

Problem 0.1 (Greek alphabet)
Fill in the blanks in the following table of Greek letters. Note that capitalized names denote capital Greek letters. 3pt

2min

Symbol			Ξ		ϖ	Γ	Λ		
Name	Phi	Psi		nu				Omega	gamma

Problem 0.2 (Binomial Coefficients)

8pt

Let us define the binomial coefficients using Pascal's formula: 10min

$$\binom{0}{0} = \binom{1}{0} = \binom{1}{1} = 1, \binom{n}{0} = \binom{n}{n} = 1, \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Prove that:

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

Hint: You can imagine the binomial coefficients in the form of Pascal's triangle. On the nth row you have all the binomial coefficients of type $\binom{n}{k}$ in increasing order of k. Each element is the sum of the two previous elements.

$$\begin{array}{c}
1 \\
1 \quad 1 \\
1 \quad 2 \quad 1 \\
1 \quad 3 \quad 3 \quad 1 \\
1 \quad 4 \quad 6 \quad 4 \quad 1 \\
1 \quad 5 \quad 10 \quad 10 \quad 5 \quad 1
\end{array}$$

Solution: The base case is already given in the definition.

Assume that, for some n the statement holds.

For $n+1$ we notice that the sum of all elements of the nth row is twice the sum of the elements in the previous row (the 1s at the end of the previous row are summed up only once but there are two more 1s in the nth row). Therefore the statement holds.

---

2 Relations and Functions

Problem 0.3 (Properties of Relations)
Suppose that R and S are reflexive relations on a set A. Prove or refute by a counter-example each of these statements: 8pt

10min

1. $R \cap S$ is reflexive
2. $R \cup S$ is reflexive
3. $S \circ R$ is irreflexive
4. $R \backslash S$ is irreflexive

Problem 0.4 (More proofs)

Prove or refute the following statement. If you want to refute it, a counterexample is enough. 4pt

6min

1. function $f\colon \mathbb{R}^+ \to \mathbb{R}^+; n \mapsto \frac{\sqrt{x^2-4}}{\sqrt{x-2}}$ is total.
2. function $f\colon \mathbb{N} \to \mathbb{N}; n \mapsto \lfloor \frac{n!}{2^n} \rfloor$ is surjective.

3 Abstract Data Types and Abstract Procedures

Problem 0.5 (Shop-aholic)

I am a shop-aholic, and I want to represent my online purchases with an ADT. 8pt

8min

Every order I submit online should contain a list of items I want to buy, my delivery address, my account number and whether it is rush delivery (€6 additional fee if I want my purchase arrive on campus the next day). Each item in the list should have a name, price, and quantity.

Think of what sorts (datatypes) you can use to represent all the information.

Note: You can assume the ADTs for real numbers and strings as given, they come with pre-defined functions for equality. You can also use addition and multiplication for reals, but no more.

---

1. Design an ADT for my order. Now you want to buy one iPhone6+ and one iPad mini, and want to have them delivered to campus tomorrow. Represent your purchase using your ADT.

   Note: Do not disclose your account number, just make one up if you want.

   ---

2. Design an abstract procedure that deletes an item from the list if the quantity of that item is zero.
3. Design an abstract procedure which returns the amount I need to pay for my order.

Hint: The price of iPhone 6 plus with 16 GB is €819, iPad mini2 with 16 GB is €313.44.

Problem 0.6 Given the abstract data type 3pt

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

4min

Which of the following mappings are substitutions? Justify in the cases where they are not.

- $\sigma_1 := [f(x_{\mathbb{A}})/x_{\mathbb{A}}], [b/y_{\mathbb{B}}]$
- $\sigma_2 := [g(a,b)/x_{\mathbb{A}}], [g(a,b)/y_{\mathbb{B}}], [f(x_{\mathbb{A}})/z_{\mathbb{A}}]$
- $\sigma_3 := [f^{i+1}(x_{\mathbb{A}})/f^i(x_{\mathbb{A}})]$, $i \in \mathbb{N}$ where $f^0(x_{\mathbb{A}}) = x_{\mathbb{A}}$ and $f^{i+1}(x_{\mathbb{A}}) = f(f^i(x_{\mathbb{A}}))$

Note: $f^k(x)$ here just means f applied to x for k times consecutively, i. e. $\underbrace{f(f(\ldots f}_{k \text{ times}}(x)))$.

Solution:

- substitution, as it's a mapping from $\mathcal{V}$ to $\mathcal{T}(\mathcal{A};\mathcal{V}))$
- no substitution, as it does not preserve sorts for $x_{\mathbb{A}} \in \mathcal{V}$, $g \in \mathcal{T}_{\mathbb{A}\times\mathbb{B}\to\mathbb{B}}(\mathcal{A};\mathcal{V}))$
- no substitution, as it's not a mapping from $\mathcal{V}$ to $\mathcal{T}(\mathcal{A};\mathcal{V})$

Grading: one point per part

---

4 Programming in Standard ML

Problem 0.7 (Counting words)

8pt
8min

Write an SML function **count** that, given the alphabet $\{0, 1, 2\}$ counts the number of words of length n that have no 2 0s next to each other.

```
val count = fn:int->int;
- count(2);
val it = 8:int; (*01, 02, 10, 11, 12, 20, 21, 22*)
- count(3);
val it = 22:int;
```

Hint: Think of all the possible starting string prefixes and how you can use the count of the remainder.

Solution: To construct a string of length n with no two consecutive zeros we can:

- Start with a 1 or a 2. Then we need to count how many strings of length $n-1$ there are
- Start with a 01 or a 02. Then we need to count how many strings of length $n-2$ there are

```
fun count(0) = 1
  |count(1) = 3
  |count(n) = 2*(count(n-1) + count(n-2));
```

Problem 0.8 (Programming with effects)

8pt
12min

1. Write down two SML functions that compute the following sequences. Make sure that the user does not try to access a negative index (i.e. $n < 0$) or the square root takes a negative argument by raising exceptions.

$$a_n := \begin{cases} 2 & \text{if } n = 0 \\ \sqrt{b_{n-1}} + a_{n-1} & \text{else} \end{cases} \qquad b_n := \begin{cases} 1 & \text{if } n = 0 \\ 3a_{n-1} - b_{n-1} & \text{else} \end{cases}$$

   Note: The two sequences are mutually recursive, i.e. they call each other.

   Hint: You can use the library function `Math.sqrt` for computing the square root.

2. Explain what would happen if you would not use exceptions in your functions from the point of view of termination.

Solution:

1.
```
exception InvalidIndex;

exception NegativeParameter;

fun a(0) = 2.0
  | a(n) = if n<0 then raise InvalidIndex else
           if b(n-1)<0.0 then raise NegativeParameter else Math.sqrt(b(n-1))+a(n-1)
and b(0) = 1.0
  | b(n) = if n<0 then raise InvalidIndex else 3.0 * a(n-1) - b(n-1);
```

2. For negative values, the function will never terminate, because the recursion steps will take a negative number to its predecessor, thus resulting in an infinite chain: $-1, -2, -3, \ldots$.

5 Formal Languages

Problem 0.9 (Playing with Letters)

Given the alphabet $A = \{a, b\}$ and a $L = \bigcup_{i=1}^{\infty} L_i$ where 8pt

8min

1. $L_1 = \{\epsilon\}$
2. $L_n = \{S_1 a S_2 a S_3 b S_4 \mid S_i \in \bigcup_{j=1}^{n-1} L_j, 1 \leq i \leq 4\}$

1. Determine whether the following strings are in the language or not.
   (a) $s_1 = aab$
   (b) $s_2 = aabaab$
   (c) $s_3 = aabbba$
   (d) $s_4 = abbababbabbaa$ (hint: length of strings in L)

   Justify your answers by arguing about the values of the S_i.
2. Modify the language such that it contains all words which have exactly twice as many letter "a"s as "b"s, like *aab*, *aaaabb*, *bbaaaa*, …