

Midterm Exam

General CS 1 (320101)

October 19, 2009

LAST NAME(s):

FIRST NAME(s):

MATRICULATION NUMBER:

Write the solutions to the sheet.

You can reach 45 points if you solve all problems. You will only need 40 points for a perfect score, i. e. 5 points are bonus points.

You have 65 minutes, so take it slow and avoid rushing to mistakes!

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	Sum	grade
total	3	5	5	5	8	5	10	4	40	
reached										

1 Mathematical Foundations

3pt

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

Symbol					γ	Σ	π	Φ
Name	alpha	eta	lambda	iota				

5pt

Problem 1.2 (Properties of Relations)

You are given the following definitions:

An irreflexive relation is a relation in which no element is related to itself.

An asymmetric relation R on a set S is a relation which has no symmetric pairs. If $\langle a, b \rangle \in R$ there exists no $\langle b, a \rangle \in R$, $\forall a, b \in S$.

Your tasks:

- Find a symmetric, an asymmetric and an antisymmetric relation on the set $\{1, 2, 3, 4, 5\}$.
- Prove that a relation is asymmetric if and only if it is both antisymmetric and irreflexive.

Problem 1.3 (Induction on Addition Chains)

Following is the definition of Addition Chain:

The series $(a_0, a_1, a_2, \dots, a_l)$, where $1 = a_0 < a_1 < a_2 < \dots < a_l = n$ is called an addition chain for $n \in \mathbb{N}$ if and only if $\forall i \in \mathbb{N} \ 1 \leq i \leq l$, $\exists j, k \in \mathbb{N}$ such that $0 \leq j, k < i$ and $a_i = a_j + a_k$.

We will call l the length of the chain.

For example: $(1, 2, 3, 5, 10, 15)$ is an addition chain for $n = 15$ of length $l = 5$.
 $(1, 2, 3, 4, 7, 10, 14, 15)$ would also be an addition chain for $n = 15$ with length $l = 7$.

Your Task:

Prove by induction that the shortest addition chain for $n \in \mathbb{N}$ meets the condition $a_k \leq 2^k \ \forall k \in \mathbb{N}, k \leq n$.

2 Abstract Data Types and Abstract Procedures

5pt

Problem 2.1 (ADT for sentences)

A *sentence* is a (possibly empty) list of **non-empty** words over the alphabet $\{a, b, \dots, z\}$.

Create an Abstract Data Type for *sentences*. Start with the constructor declarations $[a : Letter], [b : Letter], \dots, [z : Letter]$. Make sure that any sentence has a *unique* representation in your ADT.

Don't forget that the words are non-empty, but the sentence can be empty.

Problem 2.2 (Addition on integers)

In the General Computer Science Exam, John was given the following abstract datatype for extended natural numbers.

$$\langle \{N\}, \{[o : N], [s : N \rightarrow N], [n : N \rightarrow N]\} \rangle,$$

Where n is the negative of a natural number and s is the successor operation. He was then asked to come up with an abstract procedure for simple addition on all possible combination of numbers present in this datatype.

John gave the following solution:

$$\{ \text{add}(x,o) \rightarrow x, \text{add}(o,x) \rightarrow s(x), \text{add}(x,s(y)) \rightarrow s(\text{add}(x,y)), \text{add}(s(x),n(s(y))) \rightarrow \text{add}(x,n(y)), \\ \text{add}(n(s(x)),s(y)) \rightarrow \text{add}(n(x),y), \text{add}(x,o) \rightarrow x, \text{add}(x,o) \rightarrow x \}$$

He was awarded 3 out of 6 points for this answer.

Please provide all the possible errors in his solution and also all other reasons for this point deduction.

Hint: Hint : Verify the correctness, completeness and accuracy of his solution

Also give the complete and correct version of this procedure and then,

Analyze its Termination Properties.

Represent numbers 4 and -2 in this datatype and compute their sum using the given procedure (show every step of the computation).

This page was intentionally left blank for extra space

3 Programming in Standard ML

5pt

Problem 3.1 (SML types)

Determine the signatures of the following SML functions:

```
fun f1 (a,b,c,d) = [a]@(b::c)@d;  
fun f2 (a,c) = fn (l) => l(a(c)) + a(c);  
fun f3 (a)(b,c) = c(foldl(a)(b));
```

An example signature is:

```
- fun example(n,m) = (n,m);  
val example = fn : 'a * 'b -> 'a * 'b
```

Hint: The signature of `foldl` is:

```
val foldl = fn : ('a * 'b -> 'b) -> 'b -> 'a list -> 'b
```

Problem 3.2 (Another Number 4)

10pt

It is known that each natural number can be obtained from number 4 and applying the following operations:

- A - add 4 at the end of the number
- B - add 0 at the end of the number
- C - the current number is divided by 2

A GenCS student coded the room numbers of his colleagues using sequences of characters (A, B, C), which are representations of the operations needed to obtain the number from 4. Notice how these numbers are built (mind the operation order):

$$120 \rightsquigarrow B(C(A(C(4)))) \rightsquigarrow \text{"BCAC"}: \begin{cases} 4/2 = 2 \rightsquigarrow C \\ 2 \leftarrow 4 = 24 \rightsquigarrow A \\ 24/2 = 12 \rightsquigarrow C \\ 12 \leftarrow 0 = 120 \rightsquigarrow B \end{cases}$$

Your **task** to help your friends by decoding their room numbers.

1. Write a SML function *decode* that takes a list of tuples containing the name of the student and his/her coded room number. Your function should output the updated list containing names and correct room numbers. If a coded string contains other characters than the ones specified, raise exception `Undefined`.

```
val decode = fn : (string * string) list -> (string * int) list
```

2. Create a *lookup* function where, given a name and the initial list, displays his/her correct room number. If the name is not in the list, raise exception `NotFound`.

```
val lookup = fn : (string * string) list -> string -> int
```

Note: You can use functions from previous parts in subsequent parts without defining them. Examples:

```
val l = [("Tom", "CCAB"), ("Werner", "ABC)];
decode l;
val it = [("Tom",101),("Werner",204)] : (string * int) list
```

```
val l1 = [("Lena", "ACAXX)];
decode l1;
uncaught exception Undefined
```

```
lookup l "Tom";
val it = 101 : int
```

```
lookup l "Jerry";
uncaught exception NotFound
```


This page was intentionally left blank for extra space

4 Formal Languages

4pt

Problem 4.1 (Lexical Order)

Let $A := \{c, o, m, p, u, t, e, r\}$ be an alphabet and $L := \{cute, more, rum\}$,
 $M := \{cut, root, etc\}$ formal languages in A . Let $<$ with $r < e < t < u < p < m < o < c$
be a lexical order on A . Order the words in $L \cup M \cup LM$ in the lexical order $<_{lex}$ induced
by $<$.

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space