Name: Matriculation Number:

Midterm Exam
General CS 1 (320101)

October 30, 2007

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

The estimated time for solving this exam is 0 minutes, leaving you 60 minutes for revising your exam.

You can reach 0 points if you solve all problems. You will only need 25 points for a perfect score, i.e. -25 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		

Good luck to all students who take this test

1 Mathematical Foundations

2pt

Problem 1.1: Fill in the blanks in the table of Greek letters. Note that capitalized names denote capital Greek letters. 3min

Symbol	Δ	χ	Ψ	κ				
Name					Omega	mu	Theta	omicron

Solution:

Symbol	Δ	χ	Ψ	κ	Ω	μ	Θ	o
Name	Delta	chi	Psi	kappa	Omega	mu	Theta	omicron

Problem 1.2 (Properties of Relations) 3pt 9min

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

1. If R is reflexive then R^{-1} is reflexive
2. If R and S are transitive then $R \cup S$ is transitive
3. If R is symmetric then all subsets of R are symmetric
4. $R \cup R^{-1}$ is symmetric.

Solution:

1. R is reflexive implies that for all $a \in A$ we have aRa. Since $R^{-1} = \{\langle x, y\rangle \mid \langle y, x\rangle \in R\}$ it follows that for all $a \in A$ we have $aR^{-1}a$.
2. Consider the following counterexample:

   Take $a, b, c \in A$ arbitrary and distinct and let $R = \{\langle a, b\rangle\}$, $S = \{\langle b, c\rangle\}$. Then, although both R and S are transitive their union is not.
3. Consider the following counterexample:

   Take $a, b \in A$ arbitrary and distinct. Then, although $R = \{\langle a, b\rangle, \langle b, a\rangle\}$ is symmetric its subset $\{\langle a, b\rangle\}$ for example is not symmetric.
4. If aRb then $bR^{-1}a$ thus for all $a, b \in A$ such that $\langle a, b\rangle \in (R \cup R^{-1})$ we also have $\langle b, a\rangle \in (R \cup R^{-1})$. So the affirmation is true.

Problem 1.3 (An induction with fractions) 3pt 6min

Using induction, prove that for all $n \in (\mathbb{N}\backslash\{0\})$, the following equivalence holds:

$$u_n := \sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$$

.

Hint: Start by writing u_{n+1} in terms of u_n. Use the binomial formula $n+1^2 = n^2 + 2n + 1$.

Solution: Grading: 1 point for the base case, 2 for the step.

Note: once a student complained about this exercise because it required too much non-CS knowledge of math.

2 Abstract Data Types and Abstract Procedures

Problem 2.1 (ADT for rational numbers) 4pt 8min

Define an abstract data type for rational numbers. Write the numbers $\frac{2}{3}$ and $-\frac{1}{2}$ using your definition. Also define all other data types you need to construct rationals.

Hint: Recall that rational numbers are defined as the fractions $\frac{p}{q}$ where p is any integer and q is a positive integer.

Note: Make sure that there is only one representation for the integer 0!

A proper data type for rational numbers would not be allowed to have more than one representation for one value. As reducing fractions is a bit out of scope here, you can ignore this problem and permit e. g. two distinct representations for $\frac{2}{4}$ and $\frac{1}{2}$, although these rational numbers are actually equal.

Solution: Thanks to Dimitar "Dasenov" Asenov for contributing this exercise. Thanks to Felix Schlesinger for contributing to the disclaimer about mathematical soundness.

$$\langle\{\mathbb{P},\mathbb{I},\mathbb{Q}\},\{[one\colon\mathbb{P}],[suc\colon\mathbb{P}\to\mathbb{P}],[zero\colon\mathbb{I}],[pos\colon\mathbb{P}\to\mathbb{I}],[neg\colon\mathbb{P}\to\mathbb{I}],[q\colon\mathbb{I}\times\mathbb{P}\to\mathbb{Q}]\}\rangle$$

…where $\mathbb{P}$ are the positive integers, $\mathbb{I}$ all integers and $\mathbb{Q}$ rational numbers.

There are actually many ways to define this, but in any case it's important that 0 and any number $\neq 0$ are of different sorts.

$$\begin{aligned}\tfrac{2}{3} &= q(pos(suc(one)), suc(suc(one)))\\ -\tfrac{1}{2} &= q(neg(one), suc(one))\end{aligned}$$

Grading for 100% = 4 points:

- 0.5 points for each example (they're easy once the ADT works)
- 1 point per sort: naturals, integers, fractions

Problem 2.2: Given the abstract data type 3pt 7min

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

Which of the following mappings are substitutions?

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

---

3 Programming in Standard ML

Problem 3.1 (Calculate call charges) 6pt 12min

A telephone company computes the call charges in the following way:

1. price a is paid in the first minute
2. price b is paid in the next nine minutes
3. price c is paid for the remaining time

Example: for a call duration of 14 minutes the costumer pays $a + 9b + 4c$. Call charges vary for different destinations.

Given a list of tuples containing destinations (strings) and call charges (list with 3 int numbers) write a function that computes the net cost of a call. The function takes the input in the format `charge(destination:string, duration:int, the list described above)`.

Example:

```
charge("Bremen", 15, [("Bremen", [3, 2, 1]),
                      ("Bogdanci", [5, 4, 2]),
                      ("Kolkata", [8, 7, 5])]);
```

returns `val it = 26 : int`

The functions should raise an exception for an invalid duration or an invalid destination.

Note: The function should work for arbitrary tariff lists; the one above is given as an example only!

If one destination occurs more than once in the list, you can just take the first one you find and ignore subsequent ones.

Solution: Thanks to Gordan Ristovski and Ankur Modi for inventing this exercise.

Solution by Peter Nemeth (treats 0 as invalid):

```
exception invDest;
exception invDur;

fun id(a, nil) = raise invDest
  | id(a, (x, y)::l = if (a = x) then y else id(a, l)

fun match([x1, x2, x3], b) = if (b > 10) then
    (x1 + (9 * x2) + ((b - 10) * x3))
  else
    x1 + ((b - 1) * x2)

fun charge(a, b, l) = if (b < 1) then raise invDur
  else match(id(a, l), b)
```

Grading: one half of the points for the calculation and duration case distinction; the other half for the table lookup. For 100% = 12 points, deduct half a point per missing exception.

Problem 3.2 **(Cartesian and cascading functions)** 3pt 6min

Write an SML function `curry` that takes a binary Cartesian function and returns the cascading variant, and a function `descartes` that does the inverse.

Solution:

```
fun curry (f) = (fn x => (fn y => f(x,y)));
fun descartes (f) = (fn (x,y) => f x y);
```

4 Codes

Problem 4.1 (A Morse string code) 4pt 8min

Given the alphabets $A = \{\mathtt{J}, \mathtt{A}, \mathtt{C}, \mathtt{O}, \mathtt{B}, \mathtt{S}\}$ and $B = \{., -\}$, consider the following restriction μ' of the Morse code to A:

J	.—
A	.-
C	-.-.
O	—
B	-...
S	...

Answer the following questions, always by proving, giving a counter-example, or referring to a definition or theorem in the lecture:

1. Is the code μ' a character code?
2. Is μ' a string code?
   (a) If it is not a string code, suggest a change by modifying at most one codeword (without extending B for that!) and prove that the resulting code μ' is a string code.

Solution:

1. μ' is a character code (injective)
2. μ' is not a string code. Counter-example:

```
string 1:         (  J  ) (  B  )
encoded string:   . - - - - . . .
string 2:         (A) ( O ) ( S )
```

   Or: JOCO = AOOAJ

   Note: The "justification" that it's not a string code because it's not a prefix code $(.- \lhd .---)$ is invalid, because the respective theorem only yields "prefix code" $\Rightarrow$ "string code", not the opposite direction

   (a) Make it a prefix code by encoding e. g. A as $.-..$

Grading: if 100% = 4 points, one point for character code, 1.5 for the proof that it is not a string code, 0.5 for a reasonable modification, one for the proof that the modified code is a string code.
