Quizzes for General CS I (320101) Fall 2013

Michael Kohlhase
Jacobs University Bremen
For Course Purposes Only

November 21, 2013

Contents

Quiz 1: Introductory Quiz(Given Sep. 10. 2012)

Problem 1.1 (GenCS Grading) 4pt

State the components of the overall grade of the GenCS course and discuss their intention.

Solution:

Problem 1.2 (What is an algorithm?) 8pt

What is an algorithm? Give 3 examples of algorithms and explain them (be creative and make sure that at least two of them are not on the slides!).

Solution: An algorithm is a collection of formalized rules that can be understood and executed, and that lead to a particular endpoint or result.

Quiz 2: Peano axioms and Induction(Given Sep. 16. 2013)

6pt

Problem 2.1 **(Peano's induction axiom)**
State Peano's induction axiom and discuss what it can be used for.

Solution: Peano's induction axiom: Every unary natural number possesses property P , if

- the zero has property P and
- the successor of every unary natural number that has property P also possesses property P

Peano's induction axiom is useful to prove that all natural numbers possess some property. In practice we often use the axiom to prove useful equalities that hold for all natural numbers (e.g. binomial theorem, geometric progression).

6pt

Problem 2.2 **(Zero is not one)**
Prove or refute that $s(o)$ is different from o.

Note: Please use **only** the Peano Axioms for this proof.

Solution:
Proof: We will prove the statement using the Peano axioms:

P.1 o is a unary natural number (axiom P1)

P.2 $s(o)$ is a unary natural number, and is different from o (axiom P2 and 1.)

$\square$

Quiz 3: Mathtalk and Sets(Given Sep. 23. 2013)

Problem 3.1 (Addition Definition) 2pt

Give the two basic rules that define the "addition" operation. Provide an example for each rule. Use the unary representation of numbers (e.g. o for 0, $s(o)$ for 1 and so on).

Solution: We define the addition operation procedurally (by an algorithm).
Adding zero to a number does not change it: $n \oplus 0 = n$
Example: $0 \oplus s(s(o)) = s(s(o))$

Adding m to the successor of n yields the successor of $m + n$: $n \oplus s(m) = s(n \oplus m)$
Example: $s(s(o)) \oplus s(o) = s(s(s(o)))$

Problem 3.2 (Talking about Sets) 10pt

Given the following sets

1. $A = \{a, b, c, d, e\}$
2. $B = \{d, f, h\}$
3. $C = \{d, f, g, i\}$

Define each of the following operations on sets **in math talk** and apply it to the given sets:

1. intersection:$S \cap T :=$
   e.g. $A \cap B =$
2. union: $S \cup T :=$
   e.g. $B \cup C =$
3. set difference: $S \backslash T :=$
   e.g. $A \backslash B =$
4. n-fold Cartesian product: $S_1 \times \ldots \times S_n :=$
   e.g. the size $\#(A \times B \times C) =$

Solution:

1. intersection: $P \cap Q := \{x \,|\, x \in P \wedge x \in Q\}$
   $A \cap B = \{d\}$
2. union: $P \cup Q := \{x \,|\, x \in P \vee x \in Q\}$
   $B \cup C = \{d, f, h, g, i\}$
3. set difference: $P \backslash Q := \{x \,|\, x \in P \wedge x \notin Q\}$
   $A \backslash B = \{a, b, c, e\}$
4. n-fold Cartesian product: $A_1, \ldots, A_n := \{\langle a_1, \ldots, a_n \rangle \,|\, \forall i.a_i \in A_i\}$
   $\#(A \times B \times C) = \#(A) \cdot \#(B) \cdot \#(C) = 5 \cdot 3 \cdot 4 = 60$

Quiz 4: Relations & SML Pattern Matching(Given Sep. 30. 2013)

Problem 4.1: Given $A := \{1,2,3,4\}$, $B := \{5,6,7\}$ and following relations: 4pt

$$R_1 \subseteq A \times A, \quad R_1 := \{\langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle, \langle 1,4\rangle, \langle 4,3\rangle, \langle 1,3\rangle\}$$

$$R_2 \subseteq B \times B, \quad R_2 := \{\langle 5,5\rangle, \langle 6,6\rangle, \langle 7,7\rangle, \langle 5,6\rangle, \langle 6,5\rangle, \langle 6,7\rangle, \langle 7,6\rangle\}$$

Determine for these relations whether they are reflexive, symmetric, and transitive. If they are not, give counterexamples (i.e. examples, where the given property is violated).

Solution: R_1 is only transitive. R_2 is reflexive and symmetric.

Problem 4.2 (Pattern Matching in SML) 8pt

You are typing to the SML interpreter and it replies:

```
- val unittriple = (1,1,1);
  val unittriple = (1,1,1) : int * int * int
```

1. you continue with

   ```
   - val (_,x,_) = unittriple;
   ```

   What will the system reply? Explain briefly.

2. write an SML function `fourth` that given a quadruple of reals (members of the SML type `real`) extracts the last (fourth) component.

Solution:

1. the reply is

   ```
   val x = 1 :int
   ```

2. the function is defined as

   ```
   fun fourth (q) = let val (_:real,_:real,_:real,f:real) = q in f end
   ```

Quiz 5: SML Data Types(Given Oct. 7. 2013)

12pt

Problem 5.1 (Temperatures)

You are given the following SML datatype temp that represents temperatures in Fahrenheit and Celsius.

```
datatype temp = Celsius of real | Fahrenheit of real;
```

Write an SML function find : temp list −> temp that returns the lowest temperature in a list. For instance,

```
- find([Celsius(12.0), Fahrenheit(52.0), Celsius(32.0)]);
val it = Fahrenheit 52.0 : temp
```

Note: You can use the following formula for transforming Fahrenheit into Celsius: $t_C = (t_F - 32) \cdot \frac{5}{9}$

Solution:

```
datatype temp = Celsius of real | Fahrenheit of real;

(* transform F into C to have a ``common ground'' *)
fun value(Celsius(x)) = x
  | value(Fahrenheit(x)) = (x-32.0)*5.0/9.0;

fun find([a]) = a
  | find(a::l) = let val x = find(l) in if value(a) > value(x) then x else a end;
```

Quiz 6: Abstract Data Types(Given Oct. 28. 2013)

6pt

Problem 6.1 **(Abstract Data Type for given Ground Terms)**
Suppose the expressions $f(g(a,b),c)$ and $h(g,f(a,b))$ are both ground terms.

1. Write one appropriate abstract data type for both of them.
2. Are the expressions $f(a,b,c), h(g,a)$ and $h(f,b)$ ground terms of your abstract data type too? Justify your answer.

Solution:

1. This is a possible abstract data type for the expressions $f(g(a,b),c)$ and $h(g,f(a,b))$:

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

2. $f(a,b,c)$ does not fit to that abstract data type as f has to many arguments. The expression $h(g,a)$ is a ground term as a is of the same sort as $f(a,b)$ from the ground term $h(g,f(a,b))$ and $h(f,b)$ is also a ground term as f is of the same sort as g as well as a is of the same sort as b.

Problem 6.2 (An abstract procedure) 6pt

Given the following ADT for lists of unary natural numbers

$$L := \langle \{\mathbb{L}, \mathbb{N}\}, \{[o \colon \mathbb{N}], [s \colon \mathbb{N} \to \mathbb{N}], [nil \colon \mathbb{L}], [cons \colon \mathbb{N} \times \mathbb{L} \to \mathbb{L}]\}\rangle$$

and an abstract procedure for appending two lists,

$$\langle @{::}\mathbb{L} \times \mathbb{L} \to \mathbb{L}\,;\, \{@(cons(n, L_1), L_2) \rightsquigarrow cons(n, @(L_1, L_2)), @(nil, L_2) \rightsquigarrow L_2\}\rangle$$

Provide an abstract procedure for reversing lists!

Solution: Please refer to slide 90, page 58 of the lecture notes. The definition of reverse is:

$$\langle \rho{::}\mathbb{L} \to \mathbb{L}\,;\, \{\rho(cons(a, L)) \rightsquigarrow @(\rho(L), cons(a, nil)), \rho(nil) \rightsquigarrow nil\}\rangle$$

Quiz 7: Mutual Recursion(Given Nov. 4. 2013)

12pt

Problem 7.1 **(Mutual recursion in SML)**
Implement functions for the following recursive/ mutually recursive functions:

- the Hofstadter male and female sequences:

$$\begin{aligned} male(n) &= \begin{cases} 0 & \text{if } n = 0 \\ n - female(male(n-1)) & \text{if } n > 0 \end{cases} \\ female(n) &= \begin{cases} 1 & \text{if } n = 0 \\ n - male(female(n-1)) & \text{if } n > 0 \end{cases} \end{aligned}$$

Solution:

```
(* Hofstadter male & female *)
fun female(0) = 1
  | female(n) = if n < 0 then raise Negative else n - male(female(n-1))
and male(0) = 0
  | male(n) = if n < 0 then raise Negative else n - female(male(n-1));
```

Quiz 8: String code(Given Nov. 11. 2013)

12pt

Problem 8.1 (String codes)
Given the alphabet $A := \{1, 2, \ldots, 2008, 2009\}$ and $B := \{a, b\}$:

1. Construct a character code $c\colon A \to B^+$ whose extension is a string code
2. Prove that the extension is a string code.

Note: You can use the theorems provided in class if you state them.

Solution: Let: $a_1 = 1$,$a_2 = 2$, $\ldots$, $a_{2008} = 2008$, $a_{2009} = 2009$

1. $c(a_x) := a^{[[x]]}b^{[[2009-x]]}$
2. **Proof**:

   P.1 Theorem: If c is a code with $|c(a)| = k$ for all $a \in A$ and some $k \in \mathbb{N}$, then c is a prefix code.

   P.2 $(\forall a_x \in A.|c(a)| = 2009) \Rightarrow c$ is a prefix code.

   P.3 Theorem: The extension $c'\colon A^* \to B^*$ of a prefix code $c\colon A \to B^+$ is a string code.

   P.4 c is a prefix code $\Rightarrow$ the extension of c is a string code $\square$
