Functional Programming 1 — 1DL330

Assignment 1

Lab: Friday, 9 September

Submission Deadline: 18:00, Tuesday, 13 September, 2016

This assignment should be solved in **groups of two** students. Please join a group in the *Assignment 1* group division on the Student Portal. If you have not found a group partner by the start of the lab for this assignment, please inform the lab assistant. If you would prefer to solve this assignment individually due to special circumstances, please contact our lab assistant, albert.yang@it.uu.se.

Instructions

- Start by reading the General Lab Instructions (on the Student Portal).
- Name the file containing your solutions `lab1.sml`. The file that you submit must contain valid SML code, so the answers to some of the questions need to be placed in comments: *(∗ ... ∗)*.
- Remember to follow our Coding Convention (also on the Student Portal), and provide specifications for all functions that you write.
- Make sure that your solution passes the tests in `lab1_test.sml` before you submit. A submission that does not pass these tests will get grade K. See the *Testing* section near the end of this assignment for details.

1 Reduction, Specification and Variant

Consider the following function declaration:

```
fun product n =
  if n = 1 then
    1
  else
    n * product (n-1)
```

1. Give a detailed step-by-step evaluation of `product 3`, using the same style as in the lecture slides. Every step should be a valid SML expression.
2. What does the function compute?
3. Write a specification for the function.

4. Give a variant for the function.

Hint: See the Coding Convention (on the Student Portal) about specifications and variants.

2 Currying

Consider this function declaration:

```
fun plus x y = x + y
```

- Write the function declaration as a value declaration `val plus = ...` Your declaration should be equivalent to the declaration above.
- What happens when the declaration `val foo = plus 4 5` is entered?
- What happens when the declaration `val bar = plus 4` is entered?
- Give a step-by-step evaluation of `plus 3 4`.

3 Types

Give functions with the following types:

1. `int -> int`
2. `int -> int -> int`
3. `int -> int * int`
4. `int * int -> int`
5. `int -> real -> string -> string`
6. `int * (string * string * int) -> int * string`

In each case, try to find a function that is defined for all possible input values, and where the result depends on all parameters. Name your functions funN, where N is replaced by the type number (1–6) in the list above.

4 Divisibility

2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder.

Give an SML definition for a function `lcm n` that returns the smallest positive number that is evenly divisible (i.e., divisible without remainder) by all of the numbers from 1 to n. Use auxiliary functions as appropriate.

You must decide how to handle the case when $n < 1$.

Testing

Use the file `lab1_test.sml` (from the Student Portal) to test your solution.

1. Place `lab1_test.sml` in the same directory as your solution file `lab1.sml`.
2. Start Poly/ML in that directory: e.g., `cd lab1dir; poly`.
3. Enter `use "lab1_test.sml";` at the Poly/ML prompt to run the tests.
4. Check the output for failing tests.
5. Quit Poly/ML (e.g., by typing Ctrl+D).
6. Fix all failing tests by modifing your `lab1.sml` file.
7. Repeat steps 2.-6. until all tests succeed.

The file `lab1_test.sml` does not contain any test case for the lcm function from Exercise 4 when n is equal to -1. Add such a test case to check that your lcm function handles this case. (You do *not* need to submit this test case.)

Good luck!