

Homework 1

Please hand in a printout of your code by

Wednesday, 7 February 2007

before class meets.

Extensions can *only* be granted before the deadline

(if necessary, contact the teaching staff by email in english, please.)

Guidelines

While we acknowledge that beauty is in the eye of the beholder, you should nonetheless strive for elegance in your code. Not every program which runs deserves full credit. Make sure to state invariants in comments which are sometimes implicit in the informal presentation of an exercise. If auxiliary functions are required, describe concisely what they implement. Do not reinvent wheels, and try to make your functions small and easy to understand. Use tasteful layout and avoid long winded and contorted code.

Problem 1

Consider the following SML program.

```
let
  val A
    = "IT University"
  val B
    = String.maxSize
  val C
    = #"o"
  val D
    = 65
  val E
    = Char.chr (D + 37)
  val F
    = Char.toString C ^ Char.toString E
  val G
    = A < F
  val H
    = A ^ " " ^ F ^ " Copen" ^ (if G then "hagen" else "negah")
in
  H
end
```

1. Give the types for each of the intermediate results A ... H.
2. What is the result of the computation?
3. How did SML compute that result? Argue by considering A ... H in turn, each result of the subcomputation, and exactly, what SML to derive it.

Problem 2

Consider the following SML program.

```
let
  val A : bool
      = x
  val B : bool
      = y
  val C : bool
      = if A then
          if B then true
          else false
        else
          if B then true
          else true
in
  C
end
```

1. What is the result of evaluating this program, assuming the x and y in the code above were replaced by `true`.
2. Rewrite the code in such a way that the new type of A, B, and C is `int` and no longer `bool`. The idea being that if A was `true` then it should now be 1 and if it was `false` then it should now be 0.
3. The result C of the computation in the code above corresponds in some sense to the truth of the formula “A implies B”, which is always `true`, except for the case that A is `true` and B is `false`. Can you rewrite the program from above in such a way that C is `true` if “A and not B” is, and `false` otherwise?

Problem 3

Assume, you are buying a house, and you take out a mortgage. There is a mathematical formula to compute the amount that you will owe per month. Let p the principal amount of the loan you would like to take out, i the annual interest rate (in percent, a number between 0 and 100), and let l be the duration of the loan (in years), which means in how many years you want to pay it back.

From these three values p , i , and l , we can compute the monthly interest as $j = \frac{i}{1200}$ and the runtime of the loan in months $n = 12l$. You are asked to implement the following formula

$$m = p \frac{j}{1 - (1 + j)^{-n}}$$

For your implementation, use the following outline, where you can substitute any value within bounds for p , i , and j .

```
let
  val P : real
    = p
  val I : real
    = i
  val L : int
    = j
  ...
in
  ...
end
```

Please fill in the missing You may extend the program with new `val` declarations. Make sure to state their types. The exponentiation function is called `Math.pow`. Different from other languages, you need to case integers into floating point numbers explicitly. The relevant function is called `Real.fromInt`.