General Computer Science I (320101) Fall 2011
Assignment 5: ADTs, inductive functions and SML
(Given Oct. 14., Due Oct. 27.)

10pt

Problem 5.1 (Coprime numbers)
Two integers are called coprime if their greatest common divisor is 1.

Prove that given two coprime integers $a > 1$ and $b > 1$, there exist two integers $x < b$ and $y < a$, such that $ax - by = 1$.

10pt

Problem 5.2 (Coprime numbers in SML)
Now implement what you proved in the previous problem by creating an SML function

```
val coprimeCoeff = fn : int * int -> int * int
```

which takes two integers $a > 1$ and $b > 1$, and, if a and b are coprime returns the pair of integers (x, y), such that $ax - by = 1$, and returns (a, b) otherwise. Example:

```
- coprimeCoeff(6,17);
val it = (3,1) : int * int
- coprimeCoeff(6,18);
val it = (6,18) : int * int
```

25pt

Problem 5.3 (Sort the mushroom!)
Red Riding Hood collected in her basket three types of mushrooms: red ones, green ones and yellow ones. We know the following about each type of mushroom:

- The red mushrooms have a number of spots on their cap, and the more the merrier.
- The green mushrooms are edible only when they have a black spot on their cap.
- The yellow mushrooms are tastier when they have are less dense (and we can only measure weight and volume, but we still can determine which mushroom is tastier, can't we?).

As soon as she arrived at her grandmother's house, she started to arrange the mushrooms by color (red first, then green and then yellow), and each mushroom by it's property (for red ones, she wanted increasing number of spots, for green ones, the edible should be at the right of non-edible, and for yellow ones, in increasing order of tastiness). Help her finish faster by writing an SML data type `mushroom` that can represent a mushroom and a function `sort` that, given a list of mushrooms, returns them in the order Red Riding Hood wants them.

For example, say that the girl found 2 red mushrooms (r_1 with 10 dots and r_2 with 12 dots), 1 yellow mushroom (y with weight 2 and volume 3) and 2 green mushroom (g_1 with a black dot and g_2 without). The result of calling `sort` on the list $[r_1, r_2, y, g_1, g_2]$ would be the list $[r_1, r_2, g_2, g_1, y]$.

Problem 5.4 (Football teams) 30pt

1. Design an abstract data type for a football team. The team should be represented by the actions of adding players to it and performing substitutions. For that, we have the operations:

   - *add* adds a player to the team with the following properties:
     - **last name** – a string of characters;
     - **number** – just a natural number for his/her T-shirt;
     - **position** – goalkeeper (GK), defender (DF), midfielder (MF), or forward (FW).
   - *subst:* removes a player from the current team (by taking his/her number as argument) and adds a new player in his/her place.

2. Now represent the following team of great football players, using the *add* operation:

Last Name	Number	Position
Yashin	1	GK
Carlos	6	DF
Maldini	3	DF
Cafu	2	DF
Hagi	10	MF
Matthaus	12	MF
Zidane	5	MF
Beckham	23	MF
Stoitchkov	8	FW
Ronaldo	9	FW
Pele	11	FW

Note: You can use "normal" base-10 notation to represent the players' numbers.

---

3. Now, as a good coach, substitute three of your players for three of your reserves using the *subst* operation:

Reserve Name	Number	Position
Barthez	16	GK
Moore	4	DF
Thuram	15	DF
Figo	7	MF
Rijkaard	13	MF
Mueller	17	FW
Maradona	20	FW

20pt

Problem 5.5 (A box of chocolates)

You are planning to open a chocolates factory in Bremen-Nord area. In order to better organize your factory's operations, you want to create an ADT that describes one particular chocolate. The health organizations are particularly interested in the mass of one chocolate, and its sweetness, so your ADT must describe that. However, you want to have as many other properties described by your ADT as you can come up with (i.e. the topping, quantity of cocoa etc.), in order to have a breakthrough on the market.

Write a description of your ADT and give its formal definition.

Hint: You should also include a sort for unary natural numbers, to use for example in mass.

Note: You are required to come up with at least 2 new sorts, other than mass, sweetness and unary natural numbers.

20pt

Problem 5.6 (Chocolates in SML)

Now that you have defined your abstract data type, create an SML data type that implements it. Also, create an SML function `box` that takes a list of chocolates and returns a tuple of elements, where the first element is the total mass of the x, the second element is the total sweetness of the box, and at least two other elements that measure interesting facts about the list of chocolates (i.e. the number of different toppings). Be creative!

Possible example:

```
- box( [
    chocolate(massFromNat(s(s(o))), zeroSugar, strawberry, hasMarzipan(false)),
    chocolate(massFromNat(s(o)), addSugar(zeroSugar), caramel, hasMarzipan(true))
]);
val it = (s(s(s(o))), addSugar(zeroSugar), 2, 1) : mass * sugar * int * int;
```

Explanation: the first element of the tuple is the total mass (3), the second is the quantity of sugar (1 unit), the third is the number of different toppings (2) and the last number is the number of chocolates that have marzipan.

Note: You are not allowed to use predefined SML datatypes when you define your chocolate datatype. (i.e. no `datatype mass = toMass of int` is allowed)

35pt

Problem 5.7 (Inductive Functions)

Considering the following inductive definitions of addition and multiplication on unary natural numbers:

- addition: $\alpha\,(n, o) = n$ and $\alpha\,(n, s(m)) = s(\alpha\,(n, m))$
- multiplication by 2: $\mu 2\,(o) = o$ and $\mu 2\,(s(n)) = s(s(\mu 2\,(n)))$
- multiplication by n: $\mu\,(n, o) = o$ and $\mu\,(n, s(m)) = \alpha\,(n, \mu\,(n, m))$

1. Carefully examine the following inductively defined functions on unary natural numbers f and g and determine what arithmetic operations do they represent. Explain how you got to your result.

- $f\,(o, o) = o$ and $f\,(o, m) = g\,(o, m)$ and $f\,(s(n), m) = s(s(s(g\,(n, s(m)))))$
- $g\,(o, o) = o$ and $g\,(n, o) = f\,(n, o)$ and $g\,(n, s(m)) = s(s(f\,(n, m)))$

2. Provide inductive defintions analogous to the ones given above for:

   - subtraction function δ that takes two arguments and computes $n - m$ (if $m > n$ your function should return o)
   - factorial function $fact$ that takes one argument and computes $(n)!$
   - modulus function mod that takes two arguments and computes $n\ mod\ m$
   - division by 2 function γ that takes a number and returns $n\ div\ 2$ (integer division)