

Declarative Programming, mid-term test, April 26 2006, 10:15am
time limit: 90 minutes, total score: 60
Standard ML (30 points)

In questions where the declaration of an SML function is asked, all SML functions found in the textbook and on the slides (both built-in and locally defined) may be used freely. Please refer to individual subquestions with their numbers and letters (e.g., 6.b)!

The types of the built-in functions appearing in the questions (with the exception of arithmetical and relational functions):

List.map	: ('a -> 'b) -> 'a list -> 'b list	explode	: string -> char list
List.foldl	: ('a * 'b -> 'b) -> 'b -> 'a list -> 'b	List.rev	: 'a list -> 'a list
List.filter	: ('a -> bool) -> 'a list -> 'a list	Char.isAlpha	: char -> bool
op::	: 'a * 'a list -> 'a list	ord	: char -> int
op@	: 'a list * 'a list -> 'a list	chr	: int -> char

5. All of the following independent, syntactically correct declarations have **two semantic errors** in them. Which are these? (7 points)

- (a) [op>("#a", "b"), (1, 2) <> (1, 2, 3), true = false]
- (b) (2*3 = 3+3, chr 95, ~9) = (6*1, "b", 0-5-4)
- (c) foldl op@ [] [4, 2, 6, 4, 1, 2.0]

6. What is the **value** of q after the evaluation of the following independent declarations? (7 points)

- (a) val (_::_::_::_::q) = explode "eas" @ rev ["r", "e", "t"]
- (b) val (_::q::_) = List.map Char.isAlpha (explode "4r3e2ald")
- (c) val q =
List.filter (fn (b, a) => a > b) [(7, 3*3), (1, 2), (ord #"z", ord #"A")]

7. Consider the following function definitions! (7 points)

```
fun zip (x::xs, y::ys) = (y, x) :: zip(xs, ys) | zip _ = []
fun f zs = zip(zs, tl zs)
fun g zs = map op- (f zs)
```

What is the **value** of x after the evaluation of the following independent declarations?

- (a) val x = g [~1]
- (b) val x = g [~1,1]
- (c) val x = g [1,3,6,10,15]
- (d) List.filter op> (f [1,4,2,3,0])
- (e) map op+ (List.filter op< (f [0,3,2,4,1]))

8. We call three neighboring elements of an integer list a sum triplet resp. difference triplet, if the sum resp. difference of the first and the third elements is equal to the middle. Write an SML function called sumdiff, which returns true if and only if the list provided in its argument contains a sum or a difference triplet. You may define auxiliary functions if you write declarative head comments for them. (9 points)

```
(* sumdiff : int list -> bool
   sumdiff zs = true iff zs contains sum or difference triplets
*)
```

```
Examples: sumdiff [1,2,4] = false;
          sumdiff [1,2,~1] = true;
          sumdiff [1,2,3,~1] = true;
          sumdiff [1,2,3,~1,2] = true;
          sumdiff [1,2,4,~3,4] = false;
          sumdiff [1,1] = false;
          sumdiff [] = false;
```