

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

In questions where the definition of a Prolog predicate is asked, all predicates 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., 2.b)!

1. Determine the outcome of the following Prolog queries (error, failure, success)! In case of success, specify the resulting variable substitutions! All queries are fed to the system independently. (5 points)

- (a) $4 * 2 = 8$.
- (b) $[a, b] = [X | Y]$.
- (c) $U + V = 5 + 7 + 2$.
- (d) $2 * 3$ is $X * Y$.
- (e) A is $2 * 4$, $B = A + 1$.

2. Write down the canonical form or draw the tree form of the both left and right hand sides of the following unifications. Specify the variable substitutions which the unifications lead to. (9 points)

- (a) $[X, a/X+b, Y+Z] = .(c, [U, U])$.
- (b) $g([2*3|L], I*b) = g([W*V, V], a*J)$.

3. Assume that the following program is loaded into the Prolog system.

```
q(X, 0) :- X >= 0, X <= 100.
q(X, 1) :- X > 100.

p([X|_], Y) :- q(X, Y).
p([_|L], Y) :- p(L, Y).
```

Determine the values that X will take as a result of the following (independent) queries! Write down *all* solutions separated by semicolons, in the same order as the system would enumerate them! If there are no solutions, write {no}!

- (a) $p([], X)$.
- (b) $p([1], X)$.
- (c) $p([1000], X)$.
- (d) $p([1, 1000, 1], X)$.
- (e) $p([1, 10, -10, 100, -100, 1000], X)$.
- (f) Assume that in the call $p(L, X)$ to the above predicate, the L argument is a list containing positive numbers. Describe in general what X values will be generated by the Prolog system and in what order. (7 points)

4. Consider a list consisting of X-Y pairs. Write a Prolog procedure which counts those list elements for which the $X*Y$ product is less than a given N value. An efficient, tail recursive solution is appreciated, but not required. If an auxiliary procedure is deemed necessary, write a declarative head comment for it! (9 points)

```
% smaller(+L, +N, -Cnt): There are Cnt number of elements in the list L
% consisting of X-Y pairs, for which X*Y is less than N. L and N are input
% parameters, Cnt is an output parameter.
```

```
| ?- smaller([10-3], 40, Cnt).          ----> Cnt = 1 ? ; no
| ?- smaller([9-4], 13, Cnt).         ----> Cnt = 0 ? ; no
| ?- smaller([10-3,8-1,6-2], 14, Cnt). ----> Cnt = 2 ? ; no
```

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;`