

When writing a Prolog program, you may use all of the Prolog predicates which were either defined during the lectures or are built-ins. Please refer to the sub-exercises using their corresponding identifier (for example 2.b).

1. What will be the results of the following Prolog goals (error, failure, success)? In case of success, specify the values of the named variables. The goals are given to the Prolog interpreter separately and on their own. (5 points)

- (a) $a-b-c = X-Y. \Rightarrow X = a-b, Y = c$
 (b) $[A,B|C] = [1,2]. \Rightarrow A = 1, B = 2, C = []$
 (c) $X = 1+2, X = 3. \Rightarrow \text{no}$
 (d) $X = a-U, X = V-2*3. \Rightarrow V = a, U = 2*3, X = a-2*3$
 (e) $X > 2, X \text{ is } 1+2. \Rightarrow \text{error}$

2. Specify the canonical forms (or draw the corresponding tree structures) of the left and right hand sides of the following equations. In the case of named variables, give the resulting variable substitutions. (9 points)

- (a) $[X, X|Y] = [c+K, L+a*b, L]. \Rightarrow X = c+a*b, K = a*b, L = c, Y = [c]$
 $.(X, .(X, Y)) = .(+ (c, K), .(+ (L, *(a, b)), .(L, [])))$
 (b) $f(P-Q, [P, Q]) = f(2*V-5, [V*U|S]). \Rightarrow P = 2*2, Q = 5, S = [5], U = 2, V = 2$
 $f(-(P, Q), .(P, .(Q, []))) = f(-(*(2, V), 5), .(*(V, U), S))$

3. Let us assume that we have loaded the following program in the Prolog system:

```
p([X|L], X, X).
p([X|L], Y, Z) :-
    Y1 is Y+X,
    p(L, Y1, Z).
```

What will the Prolog system answer if we ask the following questions (what will be the substitutions for variable X)? Enumerate all solutions in the same order as the system would, separated by a semicolon. If there is no solution, write {no}.

- (a) $p([5], 5, X). \Rightarrow 5$
 (b) $p([5,6], 1, X). \Rightarrow 6$
 (c) $p([2,3,5,11], 1, X). \Rightarrow 3; 11$
 (d) $p([1,2,4,8,16], 1, X). \Rightarrow 1; 2; 4; 8; 16$
 (e) $p([1,4,5,10,1,21,3,45,77], 0, X). \Rightarrow 5; 10; 21; 45$

Consider the following predicate which is based on the one above:

```
% p(L, Z): Z is such an element of the list L that ...
p([X|L], Z) :- p(L, X, Z).
```

- (f) Describe the declarative meaning of predicate $p/2$, i.e. complete the sentence given above. In what order does the predicate generate the solutions?

Z is an element of L that is equal to the sum of all the preceding elements, given in the original order.

(8 points)

4. We have a list of $X - Y$ pairs where X and Y are integers. Two neighbouring pairs are called *relatives* if the sum of the elements in the first pair is equal to the sum of the elements of the second pair. Implement a **Prolog procedure** called `relatives` that takes a list of $X - Y$ pairs as input and enumerates all the relatives in it with the corresponding sum as shown in the examples below. You may not define auxiliary procedures. (8 points)

```
relatives([A-B,C-D|_], A-B, C-D, S) :-
    A+B == C+D,
    S is A+B.
relatives([_|L], A, B, S) :-
    relatives(L, A, B, S).
```

When the task is to write a function, all standard functions of SML and the functions defined during the lectures can be used. Please refer to the sub-exercises using their corresponding identifier (for example 6.b).

The types of the standard functions which appear in the tasks are the following:

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

5. There are exactly two **static semantic errors** in each of the following (independent) syntactically correct SML expressions. Which are these errors? (7 points)

- (a) ("A", ord "B"!!!char vs. string!!!, "C")=("A", 66, "C", "D")!!!triple vs. 4-tuple!!!
- (b) [[1!!!int vs. real!!!], [1.0, 2.0], 1.0::2.0!!!not a list on the right of op::!!!!]
- (c) let val a::b = explode "ab"
in [b]!!!b is a list already!!! @ a!!!not a list!!!
end

6. What is the **value** of x after evaluating the following (independent) value-definitions? (7 points)

- (a) val x = rev (map Char.isUpper (explode "aB")) \Rightarrow [true, false]
- (b) val x = (if false orelse (1-2 > ~1) andalso (1-2 < ~1) then 2 else 1) + 1 \Rightarrow 2
- (c) val x = let val (b, a) = (1, (2, [3, 4])) in b + #1(a) + hd(#2(a)) end \Rightarrow 6

7. Assume the following function definitions. (8 points)

```
fun f1 [] = [] | f1 (x::xs) = f2 (f1 xs, [x])
and fun f2 ((x::xs), ls) = x :: f2(xs, ls) | f2(_, ls) = ls;
```

(a) What is the value of x after evaluating the following (independent) value definitions?

- (a1) val x = f2([], [1]) \Rightarrow [1]
- (a2) val x = f2([3, 2], [1]) \Rightarrow [3, 2, 1]
- (a3) val x = f1 [3, 2] \Rightarrow [2, 3]
- (a4) val x = f2((f1 [3, 2]), [1]) \Rightarrow [2, 3, 1]
- (a5) val x = f1 (f2([2, 1], f1 [3, 4])) \Rightarrow [3, 4, 2, 1]
- (a6) val x = foldr (fn (a, b) => f2(a, b)) [] (map (f1 o explode) ["A", "BC"])
 \Rightarrow [#"A", #"C", #"B"]

8. We have a list of (x, y) pairs where x and y are integers. Two neighbouring pairs are called *relatives* if the sum of the elements in the first pair is equal to the sum of the elements of the second pair. Implement an **SML function** called *relatives* that takes a list of (x, y) pairs as input and returns a list of all the relatives with the corresponding sum packed in triples as shown in the examples below. You may not define auxiliary functions. (8 points)

```
fun relatives ((a,b)::(c,d)::xs) =
  if a+b = c+d then ((a,b), (c,d), a+b)::relatives ((c,d)::xs)
  else relatives ((c,d)::xs)
| relatives _ = []
```