

Declarative Programming. Sample mid-course test.

Time: 45 minutes. Total score: 20 points

This is “closed book” test. You must not use the computer during the test. However, you may ask the instructor for clarification or hints, if needed.

When writing down solutions, please refer to the (sub)questions using their numeric (and alphabetic) label, e.g., 2.b.

In questions where you are asked to write a Prolog predicate, you may use all built-in predicates as well as those found in the slides.

1. Determine whether the execution of the following Prolog queries results in an *error*, *failure*, or *success*. In case of success, specify the variable substitutions of named (non-void, i.e. non `_`) variables. All queries are fed to the system independently, i.e. typed on their own after the `| ?-` prompt. (6*0.5 = 3 p)

- (a) `\+ X = 6, X = 2.`
- (b) `[a|[b,c,d]] = [X,Y|L].`
- (c) `A*B = 2*5*(7+2).`
- (d) `2+3 is U+V.`
- (e) `N is 2*4, M = N+1.`
- (f) `append([2|R], S, [3,4,5]).`

2. Consider the following invocations of the predicate `=/2`. Write down the canonical form (or draw the tree form) of the two Prolog terms appearing on the left and right hand sides of the equation (1–1 p). Determine the variable substitutions which the unification leads to, i.e. the output produced by Prolog if the given query is fed to the system (1 p). (2*3 = 6 p)

- (a) `s(8,[9-Y|U],X) = s(Y,[X],T).`
- (b) `[.(A,[])|[g(C),n+A]] = .([p],.(g(m+B),[B])).`

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

```
p([_|Xs], A, Y) :-
    A1 is A+1,
    p(Xs, A1, Y).
p([X|Xs], X, X)
```

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}`. (4*1+1 = 5 p)

- (a) `p([], 5, Y).`
- (b) `p([3,2,1,3], 1, Y).`
- (c) `p([0,3,2,1], 0, Y).`
- (d) `p([1,3,9,5,6,4,8], 2, Y).`

Consider the following predicate, which builds on the predicate `p/3` defined above:

```
% p(L, Z): Z is an element of the L list such that...
p(L, Z) :- p(L, 0, Z).
```

- (e) Provide a declarative head comment for the predicate `p/2`, i.e. expand the head comment above to a full sentence. Furthermore, describe in what order are the solutions enumerated.

4. Consider a list `L` consisting of integers. We call a list element “nice”, if it is larger than the preceding element of `L` by exactly `N`, where `N` is a given integer parameter. Write a Prolog procedure which takes a list of integers, an integer parameter and returns the list of nice elements of the given list, in the order these occur in the given list. (6 p)

```
% nice_elems(+L0, +N, -L): L is a list of "nice" elements of L0, with
% respect to parameter N. L0 and N are input parameters, L is an output
% parameter.
```

```
| ?- nice_elems([1,2,3,5,6,7], 1, L).      ----> L = [2,3,6,7] ? ; no
| ?- nice_elems([1,2,3,4,4,5,5,5], 0, L).  ----> L = [4,5,5] ? ; no
| ?- nice_elems([1,2,3,4,5,6], 2, L).     ----> L = [] ? ; no
| ?- nice_elems([], 5, L).                ----> L = [] ? ; no
```