

60-100 Group Assign #9 Thursday 10th and Friday 11th of Nov 2011

Answer all questions

MAKEUP CLASS TEST # 2 – December 3rd 2010

OPERATIONS ON DATA

(1) What is the type of the program `g` defined as follows:

```
g x y z = x : z
g [2] y z = z
g x y z = z ++ z
```

(2) Write a Miranda programs to implement the following relational algebra operations:

- a) `project_second_and_third_of_3`
- b) `select_second_of_2`

(3) Let $p = \{('x', 4), ('z', 6), ('x', 3)\}$
 $q = \{('w', 3), ('z', 4), ('z', 3)\}$

What is the result of the following expression (show all intermediate results):

```
project_first_of_3
  (join_second_of_2_with_second_of_2 p (select_first_of_2 'z' q))
```

RECURSION

(4) Let `f` be defined as follows:

```
f [] n = n
f s 0 = 5
f (x:xs) n = f xs (n - 1) + f xs (n - 2)
```

what is the value of the following application `f [1, 2, 3] 8`

(5) Use **recursion** to define a program `p5` which takes a list as input and which returns the list with all duplicates removed. For example:

```
p5 [3, 5, 7, 3, 6, 7, 7] => [3, 5, 7, 6]
                           (the order does not matter)
```

You MUST use recursion. You can assume that the member program is available, such that `member n x` returns `True` if `x` is in the list `n` and `False` otherwise.

(6) Use **recursion** to define a program `p6` which takes a list and a number as input as input and which returns `True` if the number occurs an odd number of times in the list and `False` otherwise.

```
p6 [5, 8, 3, 9, 8, 2] 8 => True
p6 [6, 4, 7, 2, 4, 8, 4] 4 => False
```

You MUST use recursion. You can assume that the member program is available.
SYNTAX OF LANGUAGES

(7) Let the grammar G be defined as follows:

```

terminals      = { 'a', 's', (, ) }
start symbol   = string
non-terminals  = {string, as, ss}
production rules = {string = 'a'
                    | 's'
                    | ( as ) string
                    | string ( ss )
                    as = 'a'
                    | 'a' as
                    ss = 's'
                    | 's' ss

```

Show how the expression ('a') 's' ('s' 's') can be derived in the grammar G.

(8) Show the parse tree (i.e. the syntax tree) for the expression ('a' 'a') 's' ('s') with respect to the grammar G.

(9) Construct a grammar for the language L which contains the following expressions, as well as other similar expressions. Note that each expression is a list of lists.

```

{ [[2,3],[4]],
  [[1,2],[3,5,4]],
  [[3,7,2],[2],[4,3]], etc

```

SEMANTICS OF LANGUAGES

(10) Construct an attribute grammar for the language L so that, the result returned for each expression in the language is the sum of the products of the numbers in the inner lists. For example:

```

{ [[2,3], [4]]           => 10 (i.e. 6 + 4)
  [[1,2], [3,5,4]]       => 62 (i.e. 2 + 60)
  [[3,7,2], [2], [4,3]] => 56 (i.e. 42 + 2 + 12)

```