

60-100 Final Exam Fall 2007 – December 19th, 12 noon to 3pm

Answer ALL ten questions. SHOWALL WORK to obtain marks. Put a box around your final answer. Answer in ink if you wish to review your paper after it has been marked.

Q1)

a) Write a Miranda program `p1a` which takes a list of pairs of numbers as input and which returns the sum of each pair's product. **You must use `foldr`, `map` and `composition (.)`** in your answer. The following is an example of the use of `p1a`:

```
p1a [(3,4), (2, 5), (3,1)] => 25
```

b) Use a **list comprehension** to define a program `p1b` which takes a list of numbers as input, and which returns a list of squares of all those numbers in the input that are greater than 5. For example:

```
p1b [3, 6, 4, 7, 8] => [36, 49, 64]
```

Q2)

a) Consider the following program:

```
p2 [] = 4
p2 [x,y] = 3
p2 (x:xs) = p2 [1,2] + p2 xs
```

What is the result of the application: `p2 [3,4,5,6,7]`

b) What is the type of the following data item: `([], [(3, 'a'), (4, 's')]):[]`

Q3) What is the type of the following program `p3`:

```
p3 a b c = [c], if a!2 = 's'
          = b!0, otherwise
```

Q4)

a) Write a Miranda program to implement the following relational algebra operator:

```
select_second_of_three
```

b) Write a Miranda program to implement the following relational algebra operator:

```
join_first_of_two_with_second_of_three
```

Q5) Use **RECURSION** to write a Miranda program `p5` which takes two lists as input and which zips up the lists into a single list of pairs. For example:

```
p5 [4, 5, 2, 7] ['f', 'h', 's'] => [(4, 'f'), (5, 'h'), (2, 's')]
```

Note that if the two lists are not of the same length, then the surplus elements in the longer list are discarded.

Q6) Let the grammar G be defined as follows:

```

G = (terminals = [0, 1, 2, 3, -, +, *, ^, (, ) ]
     non-terminals = {dig, op, expr}
     distinguished non-terminal = expr
     productions = { expr ::= num
                    | expr op expr
                    | -expr
                    | ( expr )
                    dig ::= 0 | 1 | 2 | 3
                    op ::= + | * | ^ } )

```

Draw a syntax tree for the following expression: $-(2^{(3+4)})$

Q7) Write down an attribute grammar for the following language:

Example expression	Value
[]	0
[20/4]	5
[28/4, 40/5]	7.5
[10/5, 40/4, 18/6]	5
etc.	

Note that the value is the average of the values in the list. You can assume that no numerator or denominator has more than two digits. (in x/y , x is the numerator and y is the denominator).

Q8) Use structural induction on the length of the first list to prove that the length of the output from the program p8 is equal to the sum of the lengths of its two input lists: That is, prove: $\#(p8\ n\ m) = \#n + \#m$

```

p8 [] ys = ys
p8 (x:xs) ys = x (p8 xs ys)

```

Q9) Calculate and illustrate on a graph the complexity of the program p9 with respect to the length of the input list:

```

p9 [y] = [y]
p9 (x:xs) = p9 [x] ++ (x:xs) ++ p9 xs

```

Q10)

a) Use a truth table to determine if the following formula is satisfiable, universally valid, or unsatisfiable, where \sim is logical negation:

$$(x \rightarrow y) \rightarrow (z \rightarrow (\sim x \wedge (y \& z)))$$

b) Use a truth table to show that the formula $F = (x \rightarrow z)$ is or is not a logical consequence of the set of formulas $S = \{(y \wedge x), (\sim z \rightarrow \sim y)\}$