
Midterm Exam, COMP3031, Fall 2013

Date Oct 28, 2013 (Monday)

Time 19:00-21:00

Instructions: (a) This exam contains six problems, counting for a total of 100 points.
(b) Write ALL answers in the exam book. Do not use any other papers.

Name:	Problem	Points
Student ID:	1.	
ITSC Account:	2.	
	3.	
	4.	
	5.	
	6.	

Total:

Problem 1 (15 pts) What is the value of each of the following SML expressions (a)-(c)?

```
(*a*)  
fun weird(x:real) = let val x=x+x val x=x*x in x+x end;
```

```
(*b*)  
fun avg x y = (x+y) div 2;  
  
fun run avg [x] = x |  
  run avg [x,y] = avg x y |  
  run avg (x::y::t) = run avg ((avg x y)::t);
```

```
(*c*)  
datatype 'a st = st of 'a list;  
  
fun po (st L) = st(tl L);  
fun pu (st L) x = st(x::L);  
fun pr (st L) = L;
```

(a)

```
weird 3.0;
```

(b)

```
run avg [1,2,3,4,5];
```

(c)

```
pr (pu (po (po (pu (pu (pu (st nil) "a") "b") "c") )) "d");
```

Problem 2 (15 pts) What is the type of each of the following SML functions (a)-(c)?

(a)

```
(*check*)  
fun check ((x,y), (x2,y2)) = (x=x2) orelse (y=y2);
```

(b)

```
(*sync*)  
fun sync L [] = [] |  
    sync [] R = R |  
    sync (x::xs) (y::ys) = if (x>0.0) then (sync xs ys) else (y::(sync xs ys));
```

(c)

```
(*bin*)  
fun bin f L R [] = [L, R] |  
    bin f L R (head::tail) =  
        if f head  
        then bin f (head::L) R tail  
        else bin f L (head::R) tail;
```

Problem 3 (20 pts) Write the following SML functions (a)-(b).

- (a) `combi = fn : 'a list -> 'a list list`. Given a list, `combi` returns a list of two-element combinations. Order in the output list does not matter. e.g., `combi []` returns `[]`, `combi [1]` returns `[]`, and `combi [1,2,3]` returns `[[1,2],[1,3],[2,3]]`.
- (b) `top = fn: int list -> (int*int)`. Given an integer list, `top` returns a tuple consisting the element that has the highest number of occurrences in the list and the number of occurrences. If two elements have the same number of occurrences, return the element whose first occurrence precedes that of the other element in the list. Assume the input list is non-empty. E.g., `top [1,2,1,2,3,1,4,5,6]` returns `(1,3)`, and `top [2,2,1,1,3,3,4,4]` returns `(2,2)`.

Problem 4 (15 pts) Consider the following grammar in BNF with $\langle S \rangle$ being the starting non-terminal:

$$\langle S \rangle ::= \langle M \rangle 1 \mid \langle S \rangle 0 \mid 0$$
$$\langle M \rangle ::= \langle N \rangle 0 \mid \langle S \rangle 1 \mid 1$$
$$\langle N \rangle ::= \langle M \rangle 0 \mid \langle N \rangle 1$$

- (a) Generate all strings of length less than 5 in the language represented by this grammar.
- (b) Determine whether the string 101010 belongs to the language generated by the grammar. If your answer is yes, draw a parse tree for the string; If your answer is no, just say so and no explanation is needed.

Problem 5 (15 pts) Consider the following grammar in BNF with $\langle S \rangle$ being the starting non-terminal:

$\langle S \rangle ::= \langle S \rangle \langle S \rangle | ab | ba$

- (a) Is this grammar ambiguous? If your answer is yes, write an **unambiguous** grammar in BNF to represent the language; if your answer is no, just say so and no explanation is needed.
- (b) Is the language represented by the grammar a regular language? If your answer is yes, write a regular expression to represent the language; if your answer is no, just say so and no explanation is needed.

Problem 6 (20 pts) Consider the following definition of relational algebra (RA) expressions:

- Relations "R", "S", "T", "U", and "V" are RA expressions.
- Given RA expression A, "("A")" is an RA expression.
- Given RA expression A, "SELECT" A, "PROJECT" A are RA expressions.
- A "JOIN" B is an RA expression given A and B are RA expressions.
- A "DIV" B is an RA expression given A and B are RA expressions.

The RA operators obey the following rules in **decreasing precedence** (operators on the same line have the same level of precedence):

()
SELECT PROJECT (right associative)
JOIN DIV (left associative)

- Write an **unambiguous** context-free grammar in BNF for such RA expressions, preserving the precedence and associativity of the RA operators.
- Draw the **tree representation** of the following RA expression:
"PROJECT SELECT (R JOIN S DIV T) JOIN (SELECT U DIV PROJECT V)".

Blank Page