Midterm Exam, COMP3031, Fall 2016

Date	Oct 18, 2016 (Tuesday)
Time	12:00-13:20
Instructions:	(a) This exam contains five problems, counting for a total of 100 points.
	(b) Write ALL answers in the exam book. Do not use any other paper.

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

Total:

Problem 1 (10 pts) What is the value of each of the following SML expressions (a)-(b)?

```
(a)
let
fun f x = x ^ x val g = f o f in (f "1", g "2.0")
end;
```

```
(b)
fun s ([], b) = [b]
| s (h::t, b) = s (t, b) @ s (t, h::b);
fun ps (L) = s (L, []);
ps [true,false,true];
```

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

(a)

```
fun f (a, []) = []
  | f (a, (x,y)::tail) =
if a=x then y :: f(a,tail) else f(a,tail);
```

(b)

```
fun p (f, g) x = (f x, g x);
```

(c)

```
fun f g h x = (g h) x;
```

Problem 3 (30 pts) Write the following SML functions (a)-(b).

(a) `val detuple = fn : ('a * 'b) list -> 'a list * 'b list`. Given a list L of 2-tuples, the function detuple returns a tuple consisting two lists. The first list in the returned tuple consists of all of the first elements of the tuples in L in the same order as they appear in L, and the second list in the returned tuple consists of all of the second elements of the tuples in L in the same order as they appear in L. Examples:

```
- detuple [];
val it = ([],[]) : ?.X1 list * ?.X2 list

- detuple [(1,2)];
val it = ([1],[2]) : int list * int list

- detuple [(1,2), (3,4), (5, 6)];
val it = ([1,3,5],[2,4,6]) : int list * int list

- detuple [("a",4),("c",2),("e",1)];
val it = (["a","c","e"],[4,2,1]) : string list * int list
```

(b) `val split = fn : 'a list -> 'a list * 'a list`. This function splits the input list L into two output lists, and return these two output lists in a 2-tuple. The first output list contains the first, the third, the fifth, ... and all elements of L at the odd number index in the same order as they appear in L, and the second output list contains the second, the fourth, ..., and all the elements in L at the even number index, in the same order as they appear in L. Examples:

```
- split [];
val it = ([],[]) : ?.X1 list * ?.X1 list

- split [1];
val it = ([1],[]) : int list * int list

- split [1,2,3,4,5];
val it = ([1,3,5],[2,4]) : int list * int list

- split ["a","b","c","d"];
val it = (["a","c"],["b","d"]) : string list * string list
```

Problem 4 (15 pts) Consider the following grammar in BNF with `<S>` being the starting non-terminal:

```
<S>::= <S1>:<S>|<S1>
<S1>::= <V><D>|<T><D>|2<F>
<V>::= 0|1|2|3|4|5
<D>::= 0|1|2|3|4|5|6|7|8|9
<T>::= 0|1
<F>::= 0|1|2|3|4
```

(a) Determine whether the string "23:59:59" belongs to the language generated by the grammar. If your answer is yes, draw a parse tree of the string based on the grammar; If your answer is no, just say so and no explanation is needed.

(b) 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.

Problem 5 (30 pts) Consider the following definition of two-digit integer arithmetic (2DIA) expressions:

- Two-digit integers with each digit from 0 to 9, e.g., "23", "04" are 2DIA expressions.
- Given a 2DIA expression A, A"<<", A">>", and "~"A are all 2DIA expressions.
- Given two 2DIA expressions A and B, A "+" B, A "-" B, A "*" B, and A "/" B are all 2DIA expressions.

The operators of 2DIA expressions obey the following rules in **decreasing precedence** (operators on the same line have the same level of precedence):

```
* +                    (left associative)
/ -                    (right associative)
<< >>                  (left associative)
~                      (right associative)
```

(a) Write an **unambiguous** context-free grammar in BNF for such 2DIA expressions, preserving the precedence and associativity of the operators.

(b) Draw the **tree representation** of the following 2DIA expression:

```
"~09*23/45+67-18<<>>"
```

Blank Page