

## CS 3304 Midterm Exam

Name:

ID:

---

1. (20 points) Consider the grammar

$$\begin{aligned} S &\rightarrow aSbS \\ S &\rightarrow bSaS \\ S &\rightarrow \epsilon \end{aligned}$$

where  $S$  is a non-terminal,  $a, b$  are terminals, and  $\epsilon$  is the empty symbol. Show that this grammar is ambiguous.

2. (20 points) Consider the function `try` written in some ‘mystery’ language

```
function try(a,b) =  
    if a=0 return 1  
    else   return b;
```

Assume we employ `try` in the expression

```
try(0,1/0)
```

Is it advisable that mystery use normal-order evaluation or applicative-order evaluation?

3. (20 points) Write a recursive ML function `cycle` that, given a list  $L$  and an integer  $i$ , cycles  $L$   $i$  times. For instance, if  $L = [1, 4, 5, 2]$ , then `cycle(L,1)` returns  $[4, 5, 2, 1]$ . Similarly, `cycle(L,2)` returns  $[5, 2, 1, 4]$ . Assume that  $i$  is greater than or equal to zero.
4. (5 points) Mention five ‘binding times’ in the study of programming languages, and arrange them in order of earliest to latest.
5. (15 points) Write a recursive ML function `flip` that flips alternate elements of a list. That is, given a list  $[a_1, a_2, \dots, a_n]$  as arguments, produce  $[a_2, a_1, a_4, a_3, a_6, a_5, \dots]$  as the answer. If  $n$  is odd,  $a_n$  remains at the end. What is the type of `flip`?
6. (4+4=8 points) Consider the following skeletal program in a Pascal-like block structured language:

```
program main;  
    var x : integer;  
    procedure p1;  
        var x: real;  
        procedure p2;  
            begin  
                ...  
            end;  
        end;
```

```

begin
    ...
end;

procedure p3;
begin
    ... x ... (* line 1 *)
end;

begin
    ....
end.

```

If this language uses static scoping, what is the type of the variable `x` referred to in line 1? What if the language used dynamic scoping?

7. (12 points) Here are some type and variable declarations in Pascal syntax.

```

type
    range = -5..5;
    table1 = array [range] of char;
    table2 = table1;
var
    x, y: array [-5..5] of char;
    z: table1;
    w: table2;
    i: range;
    j: -5..5;

```

State which variables are type equivalent under (a) structural equivalence, (b) name equivalence, and (c) declaration equivalence.