

Name:
SSN :

CSE 428

Fall 1998

Midterm #2

18 November 1998

The exam consists of 5 problems on 5 pages, totaling 100 points. Read each question carefully and use your time judiciously.

Write your name/number on every page.

1. Give the *most general* types for each of the following function declarations.

(20 pts)

(a) `fun Curry3 f x y z = f(x,y,z);`

(b) `fun unCurry3 f (x,y,z) = f x y z;`

(c) `fun nf (f,0) (x,y) = (x,y)
|nf (f,n) (x,y) = f(nf (f,n-1) (x,y));`

(d) `fun comp nil = 0
|comp (f::L) = f(comp L);`

2. Recall the definition of iterators for Assignment 6:

(20 pts)

```
fun iter(f,v) 0 = v
  | iter(f,v) n = f(iter(f,v) (n-1), n);

fun gen_iter(f,g,h) (0,a) = g(a)
  | gen_iter(f,g,h) (n,a) = f(gen_iter(f,g,h) (n-1,h(n,a)), n,a);
```

Give the value produced by each of the expressions below.

(5 pts)

(a)

```
let val p = iter(fn(x,y)=>x+y*y,0)
  in p 3
  end;
```

(5 pts)

(b)

```
let val q = iter(fn(x,y)=>2*x,1)
  in q 4
  end;
```

(10 pts)

(c)

```
let val r = gen_iter(fn(x,y,z)=>x*z,fn(x)=>1,fn(x,y)=>y)
  in r(2,3)
  end;
```

3. Recall the following function definitions:

(20 pts)

```
fun map f nil = nil
  |map f (x::xs) = (f x)::(map f xs);
fun accumulate f a nil = a
  |accumulate f a (x::xs) = accumulate f (f a x) xs;
fun reduce f a nil = a
  |reduce f a (x::xs) = f x (reduce f a xs);
fun cons x y = x::y;
fun snoc x y = y::x;
```

For each of the following functions, give a brief explanation of what it does. (Each performs a commonly-defined function and can be described in a few words.)

(a) `fun a(ys) = accumulate snoc nil ys;`

(b) `fun b(ys,zs) = reduce cons zs ys;`

(c) `fun c(ys) = accumulate (fn x=>fn y=>x+y) 0 (map (fn z=>1) ys);`

(d) `fun d(f,ys) = reduce (fn x=>fn xs=>f(x)::xs) nil ys;`

4. Recall the datatype declaration for types:

(20 pts)

```
datatype tp = tpint | tpbool | tpVar of string | ** of (tp * tp)
           | --> of (tp * tp);
infixr 8 ** ;
infixr 4 --> ;
```

Write a function `ground: tp -> bool` which takes an object `t:tp` and returns `true` if `t` contains no type variables (i.e., no occurrences of `tpVar`).

5. Consider the following inductive definition of propositional formulae: (20 pts)

- An identifier is a formula;
- If p is a formula then $\neg p$ is a formula;
- If p_1 and p_2 are formulas, then $p_1 \wedge p_2$ and $p_1 \vee p_2$ are both formulae.

We will assume that an identifier is simply any string.

(a) Give an SML *datatype* declaration for propositional formulae. (You do not need to include infix declarations)

(b) What is the *induction principle* for formulae?

6. In the movie *Bride of Chucky*, what was the name of Chucky's bride? (0 pts)