CS 337 **Solution to Test 2** **4/5/06**

Open book and notes.

Max points = 50 Time = 50 min Do all questions.

1. (Finite State Machine)

   (a) I reproduce Figure 1.

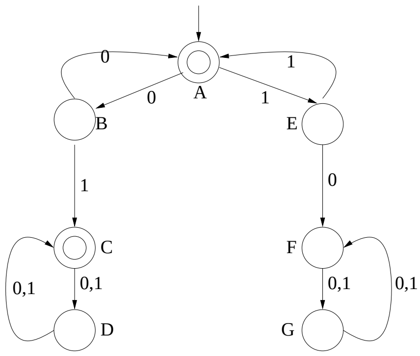

Figure 1: Finite State Machine to compare binary strings

I list the predicates for each state.

A: $f = s$
B: $f = s0$
C: $f < s$
D: $pre.f < s$
E: $f = s1$
F: $f > s$
G: $pre.f > s$

The propositions to be proven are:

$f = \epsilon,\ s = \epsilon \ \Rightarrow\ f = s$, for the initial state
(A,B): $f = s \ \Rightarrow\ f0 = s0$
(B,A): $f = s0 \ \Rightarrow\ f = s0$
(B,C): $f = s0 \ \Rightarrow\ f < s1$
(C,D): $(f < s) \ \Rightarrow\ (pre.(f0) < s)$ and $(f < s) \ \Rightarrow\ (pre.(f1) < s)$
(D,C): $(pre.f < s) \ \Rightarrow\ (f < s0)$ and $(pre.f < s) \ \Rightarrow\ (f < s1)$
(A,E): $f = s \ \Rightarrow\ f1 = s1$
(E,A): $f = s1 \ \Rightarrow\ f = s1$
(E,F): $f = s1 \ \Rightarrow\ f > s0$
(F,G): $(f > s) \ \Rightarrow\ (pre.(f0) > s)$ and $(f > s) \ \Rightarrow\ (pre.(f1) > s)$
(G,F): $(pre.f > s) \ \Rightarrow\ (f > s0)$ and $(pre.f > s) \ \Rightarrow\ (f > s1)$

   (b) Make A a rejecting state and F an accepting state.

2. (Haskell)

(a)

```
unqs []  = []
unqs [x] = [x]
unqs (x:y:xs)
 | x==y = unqs (y:xs)
 | x/=y = x: (unqs (y:xs))
```

(b)

```
ps [x] = [[x]]
ps (x:xs)  = [x]: (map (x:) (ps xs))
```

(c) $[a] \longrightarrow [[a]]$

(d) We are given

```
rev []      = []
rev (x: xs) = (rev xs) ++ [x]
```

To show `map f (rev xs) = rev (map f xs)`, we use induction on the length of `xs`.

• Base case, `xs = []`:
We have to show: `map f (rev []) = rev (map f [])`.

```
     map f (rev [])
=   {rev [] = []}
     map f []
=   {definition of map}
     []

     rev (map f [])
=   {definition of map}
     rev []
=   {rev [] = []}
     []
```

• Inductive case, `map f (rev (x:xs)) = rev (map f (x:xs))`:

```
     map f (rev (x:xs))
=   {rev (x:  xs) = (rev xs) ++ [x] }
     map f ((rev xs) ++ [x])
=   {property of map}
     (map f (rev xs)) ++ [f x]
=   {induction}
     (rev (map f xs)) ++ [f x]
=   {definition of rev}
     rev ([f x]:  (map f xs))
=   {property of map}
     rev (map f (x:xs))
```