CS 337 **Test 3** **5/10/07**

Open book and notes. Time = 105 min
Do all questions from Part 3 for Test 3 (75 points).
Bonus questions for Tests 1 and 2 appear on the last page.

1. (**PART 3**: Proofs of Recursive Programs; 15 points) Consider the following functions (from Page 126 in the book).

```
cons0 [] = []
cons0 (x:xs) = ('0':x):(cons0 xs)

cons1 [] = []
cons1 (x:xs) = ('1':x):(cons1 xs)

grayGen 0     = ([""],[""])
grayGen (n+1) = ((cons0 a) ++ (cons1 b), (cons1 a) ++ (cons0 b))
                   where (a,b) = grayGen n
```

   Prove that for all n, $n \geq 0$

```
rev a = b  where (a,b) = grayGen n
```

   Use the following facts; you don't have to prove them. For arbitrary lists `xs` and `ys`

```
rev [""]        = [""]
rev (rev xs)    = xs
rev (cons0 ys) = cons0 (rev ys)
rev (cons1 ys) = cons1 (rev ys)
rev (xs ++ ys) = (rev ys) ++ (rev xs)
```

   Show explicitly where these facts are used in your proof.

2. (**PART 3**: Higher Order Functions; 15 points)

   (a) (5 points) Define function `zip` that takes a pair of lists of equal lengths as argument and returns a list of pairs of corresponding elements. So,

   ```
   zip ([1,2,3], ['a','b','c']) = [(1,'a'), (2,'b'), (3,'c')]
   ```

   What is the type of `zip`?

   (b) (5 points) Define function `unzip` that is the inverse of `zip`:

   ```
   unzip (zip (xs,ys)) = (xs,ys)
   ```

   What is the type of `unzip`?

(c) (5 points) Define function `cross` that takes a pair of functions `(f,g)` and a pair of data `(x,y)` as input, and returns the pair `(f(x), g(y))`.
What is the type of `cross`?

3. (**PART 3**: Rabin-Karp String Matching; 11 points)

   (a) (6 points) Suppose the text string is 0110100011011101, pattern is 1101 and you are working with mod 3 in Rabin-Karp string matching. Show the positions where there is a true match and a collision. Do collisions decrease if you work with mod 5?

   (b) (5 points) Extend the Rabin-Karp method to search simultaneously for more than one pattern. Explain what generalizations need to be made.

4. (**PART 3**: String Matching; 14 points)

   (a) (4 points) Show that the core function is monotonic, that is,

   $$u \preceq v \;\Rightarrow\; c(u) \preceq c(v)$$

   (b) (4 points) Given that $u \preceq v$, is it necessarily true that $us \preceq vs$, for any symbol s? Justify your answer with a proof or a counterexample.

   (c) (6 points) The text string contains a don't-care symbol, $*$, that matches every symbol. Assume that the pattern does not contain $*$. Modify the KMP-algorithm of Page 163 to work under this additional condition. Does the core computation have to be modified?

5. (**PART 3**: Relational Databases; 20 points) You are given three relational tables: (1) table M has *worker*, *dept*, and *manager* as its attributes, where each tuple lists the name of a worker, his/her department and the name of his/her manager (a manager is also a worker); (2) table F has *worker* and *spouse* as its attributes (if the spouse is also a worker, there would be another tuple for the spouse in the table); and (3) table S has *worker* and *salary* as its attributes.

   Write queries to create the following tables.

   (a) (4 points) List of workers and their managers, where the worker salary is below $20,000.

   (b) (4 points) List of workers whose spouse is their manager.

   (c) (4 points) Average salary by department.

   (d) (8 points) Workers who are paid more than their managers. For solving this part use the following: for any table T which has attribute a, $T_{a:=b}$ is the same table in which a is renamed b and all other attributes are retained. You may rename multiple attributes as in $T_{a:=b;c:=d}$.

Bonus questions are on the next page

Bonus questions

Ypu may answer Part 1 bonus question to improve your score in Test 1 (15 points) and Part 2 bonus question to improve your score in Test 2 (15 points).

6. (**PART 1**: Error Correction; 15 points) Consider the Reed-Muller code in which the codewords are 8-bits long.

   (a) (4 points) Is every word at Hamming distance 4 from a codeword itself a Reed-Muller codeword? Justify or give a counterexample.

   (b) (5 points) Suppose the sender sends 1 0 0 1 1 0 0 1 and the receiver receives 1 1 1 1 1 0 1 1. Can the receiver detect that the transmission is erroneous? Justify your answer. If he tries to correct the errors, which codeword will he pick?

   (c) (6 points) The receiver is told that transmission of a 8-bit codeword is either completely error-free or exactly two errors are introduced in each half, left and right (so, 4 errors are introduced). With this additional knowledge, can he detect erroneous transmissions? Justify your answer.

7. (**PART 2**: Finite State Machine; 15 points)

   Draw finite state machines for the following problems.

   (a) (5 points) A machine that accepts exactly half the binary strings, i.e., for every n, $n > 0$, it accepts exactly half of all n-bit strings and rejects the other half. You are free to decide which ones are accepted.

   (b) (5 points) A machine that accepts a binary string unless 111 is a substring. So, 1011011 is accepted and 1011101 is rejected.

   (c) (5 points) A machine that receives a string of bit-pairs (x, y) as input and accepts if there are at least 3 inputs pairs (x, y) where $x < y$.